НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра автоматизованих систем обробки інформації і управління

«На правах рукопису»

УДК _____

«До захисту допущено»
В.о. завідувача кафедри
_____ *О.А.Павлов*
(підпис) (ініціали, прізвище)
“___”_____2019 р.

Магістерська дисертація

зі спеціальності *121 «Інженерія програмного забезпечення»*

на тему: «*Математичне та програмне забезпечення оцінювання захищеності інформаційних систем від веб-атак*»

Виконала: студентка VI курсу, групи *ІП-82мп*

Савчук Вікторія Володимирівна _____
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник *проф., д.т.н, доц., Стеценко І.В.* _____
(посада, науковий ступінь, , вчене звання, прізвище та ініціали) (підпис)

Консультант *доц., к.т.н., Ліщук К.І.* _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент *проф., д.т.н., доц., Клименко І.А.* _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2019 року

Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»

Факультет *інформатики та обчислювальної техніки*
(повна назва)

Кафедра *автоматизованих систем обробки інформації та управління*
(повна назва)

Рівень вищої освіти *другий (магістерський) за освітньо-професійною програмою*

Спеціальність *121 «Інженерія програмного забезпечення»*
(код і назва)

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ *О.А. Павлов*
(підпис) (ініціали, прізвище)
«___» _____ 2019р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Савчук Вікторія Володимирівна
(прізвище, ім'я, по батькові)

1. Тема дисертації *Математичне та програмне забезпечення оцінювання захищеності інформаційних систем від веб-атак*

науковий керівник дисертації *проф., д.т.н, доц., Стеценко*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від "____" _____ 20__ р. № _____

2. Строк подання студентом дисертації "____" _____ 20__ р.

3. Об'єкт дослідження *модель інформаційної системи*

4. Предмет дослідження *захищеність інформації, що обробляється в інформаційній системі*

5. Перелік завдань, які потрібно розробити *дослідити умови виконання інформаційних атак, дослідити загальну структуру інформаційної системи та методи визначення її вразливостей, розробити моделі атак інформаційної системи за допомогою мереж Петрі, розробити застосунок для моделювання хакерських атак на систему, виконати експериментальне дослідження розробленої моделі.*

6. Перелік графічного матеріалу

Модель поширення атак мережею Петрі

Модифікована модель поширення атак мережею Петрі

7. Орієнтовний перелік публікацій *Web Service Attack Simulation,*

Метод автоматизації тестування на проникнення веб-атак

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання “ *01* ” *вересня* 20 *19* р

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	*Дослідження умови виконання інформаційних атак*		
2	*Дослідження загальної структури інформаційної системи та методи визначення її вразливостей*		
3	*Постановка та формалізація математичної моделі задачі*		
4	*Розробка моделі атак інформаційної системи за допомогою мереж Петрі*		
5	*Розробка застосування для моделювання хакерських атак на систему*		
7	*Проведення експериментальних досліджень розробленої моделі*		
8	*Оформлення документації*		
9	*Подання роботи на попередній захист*	*05.12.2019*	
10	*Подання роботи на основний захист*	*16.12.2019*	

Студент _____ *В.В.Савчук*
(підпис) (ініціали, прізвище)

Науковий керівник _____ *І.В.Стеценко*
(підпис) (ініціали, прізвище)

РЕФЕРАТ

Магістерська дисертація: 93 с., 13 рис., 29 табл., 1 додаток, 43 джерел.

Актуальність. Використання в повсякденній діяльності підприємницьких компаній інформаційних систем (ІС) відкриває широкі можливості з надання комунікаційних послуг та обробки інформаційних ресурсів між відповідними відомствами, організаціями та установами, що є позитивним з точки зору перспектив для росту компаній та для їх клієнтів. Водночас існує залежність цих структур від розробників складових частин ІС, насамперед програмного забезпечення, що відкриває для зловмисників шляхи для несанкціонованого доступу до інформації, що становить реальну загрозу безпеці інформації зокрема та захищеності системі в цілому. Крім того, ймовірність реалізації відповідних загроз підвищується через широку доступність комп'ютерних технологій для всіх бажаючих. Тому проблема захисту ІС від відповідних загроз та способів їх реалізації - атак, носить надзвичайно актуальний характер.

На сьогодні є багато інструментів для тестування системи на проникність, але при ефективному скануванні системи на вразливості може бути порушена робота сервісів та вузлів, тому альтернативою в даному випадку може стати створення моделі даної системи для проведення тестів на проникнення. Створення моделі для тестування системи до стійкості атак, зменшує витрати на проведення дорогих пентестів.

Саме тому в даний час одним з найбільш актуальних напрямків досліджень в області забезпечення інформаційної безпеки є розробка методів і засобів виявлення атак і захисту від атак на інформаційні системи і мережі. В процесі такої розробки необхідно постійно проводити науково-дослідні роботи, що включають попереднє вивчення і детальне опрацювання можливих варіантів реалізації інформаційних атак. Як правило, ці роботи здійснюються з використанням моделей, що дозволяють відтворити необхідні властивості та характеристики інформаційної атаки, а також провести оцінку рівня її небезпеки для ІС. Моделі дозволяють більш точно

визначити ефективність існуючих засобів захисту за допомогою модельованих інформаційних атак.

Мета дослідження. Метою досліджень є визначення рівня вразливості сервісу та часу повного проникнення атакуючого інформаційної системи моделюючи систему за допомогою мереж Петрі.

– **Завдання дослідження.** Для досягнення мети необхідно виконати наступні завдання:

– дослідити умови виконання інформаційних атак;

– дослідити загальну структуру інформаційної системи та методи визначення її вразливостей;

– розробити моделі атак інформаційної системи за допомогою мереж Петрі;

– розробити застосунок для моделювання хакерських атак на систему;

– виконати експериментальне дослідження розробленої моделі.

Об’єкт дослідження –.модель інформаційної системи.

Предмет дослідження – захищеність інформації, що обробляється інформаційній системі.

Наукова новизна полягає в:

– застосуванні методу створення мереж Петрі для автоматизації оцінки проникності інформаційних систем;

– розширенні методів бібліотеки для візуалізації мереж Петрі.

Зв'язок роботи з науковими програмами, планами і темами.

Робота виконувалась на кафедрі автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» в рамках теми "Методи візуального програмування Петрі-об'єктних моделей". Державний реєстраційний номер 0117U000918.

Практичне значення отриманих результатів полягає у застосуванні моделі до реальних інформаційних систем для забезпечення та підтримання їх безпеки, адже на даний час захищеність системи перевіряється лише тестами на проникнення, що виконується безпосередньо людьми, з цієї причини вони є досить коштовними та задіюють ресурси апаратної та програмної частин системи.

Публікації. Матеріали роботи прийняті у вигляді статей на публікацію на конференцію “The Third International Conference on Computer Science, Engineering and Education Applications (ICCSEEA2020)” та в науковий журнал “Технічні науки та технології” Чернігівського національного технологічного університету:

- Web service attack simulation;
- Метод автоматизації тестування на проникнення веб-атак.

Master dissertation: 93 pp., 13 fig., 29 tab., 1 app., 43 sources.

The usage of information systems (IS) in the day-to-day operations of entrepreneurial companies opens enlarged opportunities for communication services provision and processing of information resources between relevant departments, organizations, and institutions, which is positive in terms of prospects for the growth of companies and their customers. At the same time, there is these structures dependence on the developers of the components of IS, primarily software. It opens the way for attackers to gain access to information that poses a real threat to the security of information and system safety as a complex. Besides, the possibility of the implementation of the relevant threat increases through the extended availability of computer technology for everyone. Therefore, the IS protection difficulty against related threats and their implementation ways (attacks) is very significant.

Currently, one of the most relevant areas of research in the field of information security is the development of methods and means of detection of attacks and protection against attacks on information systems and networks. In the process of such development, it is necessary to constantly conduct research, including preliminary study and detailed study of possible options for the implementation of information attacks. As a rule, these works are carried out using models that allow to recreate the necessary properties and characteristics of an information attack, as well as to assess the level of its danger to IS. Models allow defining more precisely the efficiency of existing means of protection using simulated information attacks.

The purpose of the research. The aim of the research is to determine the level of vulnerability of the service and the time of full penetration of the attacker of the information system by modeling the system using Petri nets.

Objectives of the search. To achieve the goal, the project must perform the following tasks:

– investigate the conditions of information attacks;

– investigate the overall structure of the information system and identify its main vulnerabilities;

– develop models of attacks of the information system by Petri nets;

– develop an application to simulate hacker attacks on the system;

– perform an experimental study of the developed model.

The object of study is a model of information system.

The subject of the research is the speed and efficiency of recognition and tracking algorithms and their synchronization.

Relationship with scientific programs, plans and themes. The work was carried out at the Department of Automated Systems for Information Processing and Management of the National Technical University of Ukraine "Kyiv Polytechnic Institute. Igor Sikorsky within the framework of the theme “Visual programming methods of Petri-object models”. State registration number is 0117U000918.

The scientific novelty of the results is:

• to apply the method of creating Petri nets to automate the estimation permeability of information systems;

• extending visualizing Petri nets library methods.

The practical value of the results obtained is to apply the model to real information systems to ensure and maintain their security, because currently the security of the system is checked only by penetration tests, which is performed directly by people, for this reason they are quite valuable and use the resources of the hardware and software parts of the system.

Publication. The materials of the work accepted as articles for publication at the conference “The Third International Conference on Computer Science, Engineering and Education Applications (ICCSEEA2020)” and in the scientific journal “TECHNICAL SCIENCES AND TECHNOLOGIES” of Chernihiv National University of Technology:

– Web service attack simulation;

– The method for automating web attack penetration testing.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

ВВ – варіанти використання

ІС – інформаційна система

ІТ – інформаційні технології

ОС – операційна система

ПЗ – програмне забезпечення

ПП – програмний продукт

ЧПВ – чиста поточна вартість

ЧДД – чистий дисконтований дохід

API – application programming interface

CVE – common vulnerabilities and exposures

CVSS – common vulnerability scoring system

CPE – common platform enumeration

DSDM – dynamic systems development method

MSF – microsoft solutions framework

NVD – national vulnerability database

RAD – rapid application development

RUP – rational unified process

РОЗДІЛ 1. СИСТЕМОТЕХНІЧНІ АСПЕКТИ ПРОЦЕСУ МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ АТАКИ

1.1 Зміст інформаційної атаки

Напад на інформаційну систему – це дія чи послідовність пов’язаних з ними дій зловмисника, які призводять до реалізації загрози шляхом використання уразливості системи. Уразливість зазвичай розуміється як слабкий момент ІС, на основі якого можлива успішна реалізація загрози. Зі свого боку, загроза - це потенційно можлива подія, дія, явище чи процес. які можуть завдати шкоди системному ресурсу [4].

Таким чином, для здійснення атаки зловмисник впливає на певну дію, яка призводить до очікуваного результату, використовуючи деякі засоби, що експлуатують вразливості системи. Інформаційна атака зазвичай може складатися з трьох етапів [5]:

а) збір інформації є основним етапом. На цьому етапі вибирається мета атаки, збирається інформація про неї (ОС, конфігурація, послуги), визначаються найбільш вразливі місця атакованої системи, вплив на які призводить до бажаного результату, тип атаки вибрано.

б) етап здійснення нападу. На цьому етапі зловмисник отримує недозволений доступ - доступ до ресурсів тих вузлів ІС, на які здійснюється атака. Якщо характер впливу нападу активний [6], то цей етап також завершується реалізацією цілей, заради яких здійснено напад. Результатом таких дій може бути порушення конфіденційності, цілісності та доступності інформації. Крім того, на цьому етапі може бути приховане джерело та факт нападу, так зване «прикриття слідів».

в) етап подальшого поширення атаки – дії, спрямовані на продовження атаки на ресурси інших вузлів ІС. У випадку пасивних атак [6] ця стадія є стадією

завершення атаки. На рисунку 1.1 показані етапи життєвого циклу типової інформації, схематично представлені інформаційною атакою.

Рисунок 1.1 - Життєвий цикл типової інформаційної атаки

При моделюванні інформаційної атаки необхідно визначити її параметри та характеристики. Основними параметрами атаки є:

- час дії;
- кратність;
- метод доступу;
- перелік вразливостей, які використовує атака;
- тип атаки;
- вплив на ціль нападу.

1.2 Основна теорія графів

Формальне визначення графа [1]:

Граф G = (V, E, φ) складається з непустого множини V, званого набором вузлів графа, множина E називається множиною ребер графа, а відображення φ від набору ребер E до набору пар елементів V[2].

Край в графі може бути спрямований, тобто пара вузлів впорядкована (з анотацією стріла). Якщо всі ребра графа спрямовані, то сам графік є спрямованим графом або орграф. Мультиграф – це граф з кількома ребрами, тобто ребрами, які мають однакові кінцеві вузли. Двосторонній граф означає, що він має два типи вузлів. Властивість дводольного графа полягає в тому, що ребро може з'єднувати тільки два вузли, які належать до різних типів.

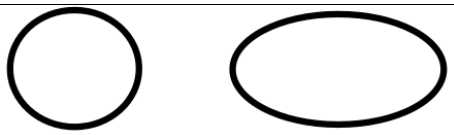
1.3 Мережі Петрі

Мережа Петрі [1, 3], яка також називається місцем або переходом [4], є узагальненим графом, введеним Карлом Адамом Петрі в 1962 році [5]. Причиною для створення мереж Петрі була необхідність вирішення проблем паралельності в системах.

Мережа Петрі – це двосторонній мультиграф, де два типи вузлів є місцями і переходами. У графічних представленнях місця показуються як кола або еліпси, а переходи – як смуги або прямокутники. У таблиці 1 представлені позначення.

Краї мережі Петрі називаються дугами і завжди спрямовані. Дуги підрозділяються на вхідні дуги, які з'єднують місця з переходами і вихідними дугами, які починаються при переході і закінчуються в місці. Місця також часто називають місцями введення та виведення, коли обговорюється певний перехід.

Таблиця 1.1 - Основні позначення мереж Петрі

Позиція	
Перехід	
Маркер	

Стан мережі Петрі позначається маркерами. Чорні точки (токени) розміщуються в місці для позначення станів. На рисунку 1.2 показана проста мережа Петрі з двома місцями, з'єднаними одним переходом і з трьома маркерами в одному з місць. Типи мереж Петрі можна відрізнити за рівнем інформації, пов'язаної з окремими маркерами, які можуть варіюватися від простої булевої інформації до структурованих маркерів. Для моделювання динамічної поведінки системи стан або маркування в мережі Петрі змінюються відповідно до правил переходу. По-перше, перехід вважається включеним, якщо кожне вхідне місце переходу позначено щонайменше одним маркером. По-друге, включений перехід може або не може спрацьовувати залежно від того, вирішується подія, чи перехід, насправді відбувся. Нарешті, випадок з включеним переходом видаляє кількість токенів з кожного вхідного місця включеного переходу і додає до кожного місця виходу переходу кількість маркерів.

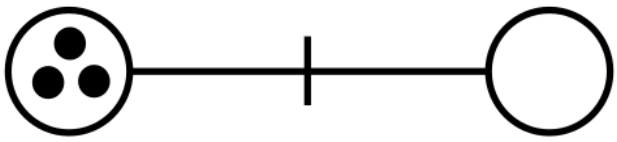


Рисунок 1.2 - Приклад мережі Петрі

Формальне визначення графіка мережі Петрі[3] полягає в наступному:

Графік мережі Петрі G є двостороннім спрямованим мультиграфом,

$G = (V, A)$, *де* $V = v_1, v_2, v_3, ..., v_n$ - множина вершин,

$A = a_1, a_2, a_3, ..., a_n$ – множина спрямованих дуг, $a_i = (v_j, v_k)$, де $v_j, v_k \in V$.

Множина V може бути розділена на дві непересічні множини P і T такі, що V = PUT, P ∩T = ∅, і для кожної спрямованої дуги $a_i \in A$, якщо $a_i = (v_j, v_k)$, *то або* $v_j \in P$ *і* $v_k \in T$ *або* $v_j \in T$ *та* $v_k \in P$.

На рисунку 1.3 представлений простий приклад мережі Петрі. Мережа на рисунку 1.3 містить по два переходи на атаку. Переходи Internet, яким передує позиція hacker, показують вихід хакеру в Інтернет. Переходи service_1 та service_2 оточені позиціями vulnerability_1, vulnerability_2 і target_host. Позиції vulnerability_1 та vulnerability_2 показують ініціацію атаки використовуючи певну вразливість сервісу, переходи service_1 і service_2 показують експлуатацію сервісу після якого хакер опиняється всередині цільового хоста.

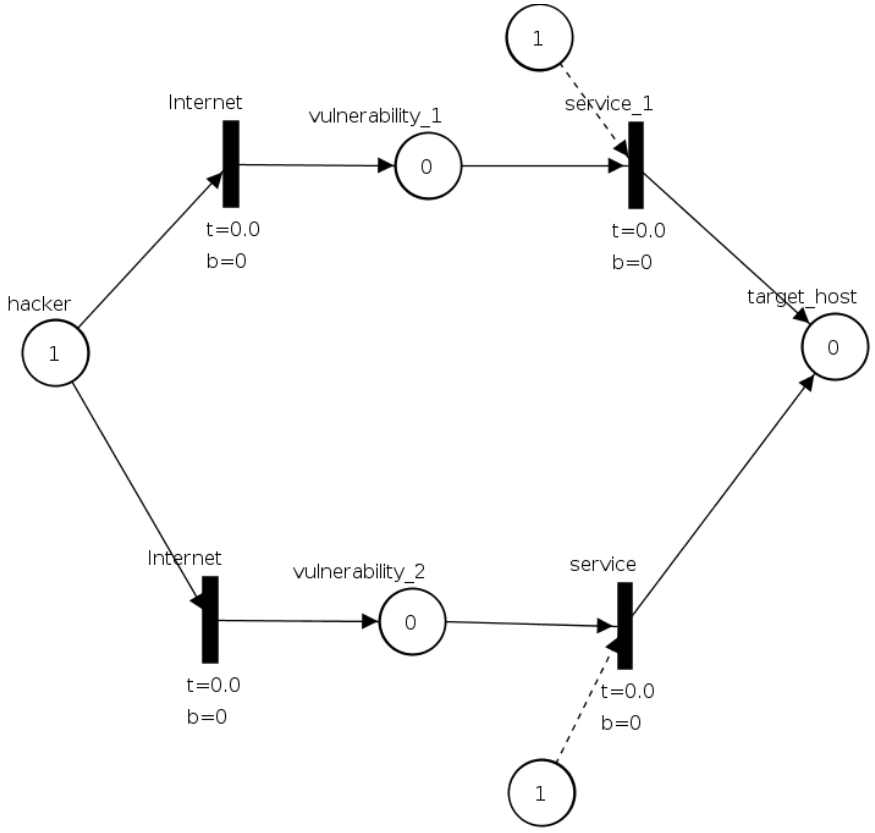


Рисунок 1.3 - Фрагмент мережі Петрі, що моделює проникнення хакеру

1.4 Сучасні методології розробки програмних засобів

Розробка ПЗ, як і інші традиційні інженерні напрямки діяльності, потребує врахування проблем якості, вартості та надійності, розв'язання яких передбачає методологія програмування.

Методологія програмування – сукупність методів, застосовуваних на різних стадіях життєвого циклу програмного забезпечення, що мають спільний філософський підхід та дозволяють забезпечити найкращу ефективність [4]. Представлене означення може бути доповнене таким чином: методологія – це реалізація стандарту; самі стандарти лише говорять про те, що повинно бути, залишаючи свободу вибору та адаптації [5]. Відповідно, під поняттям «життєвий цикл програмного забезпечення» ми розуміємо сукупність окремих етапів робіт, що проводяться у заданому порядку протягом періоду часу, який починається з вирішення питання про розроблення ПЗ і закінчується припиненням використання ПЗ [6]. Поняття ЖЦ ПЗ належить до дисципліни «Програмна інженерія». Методології являють собою ядро теорії управління розробкою ПЗ. Існують різні класифікації методологій: в залежності від використовуваної в ній моделі ЖЦ; прогнозовані та адаптивні методології.

Прогнозовані методології фокусуються на детальному плануванні майбутнього. При цьому відомі заплановані завдання і ресурси на весь термін проекту. План оптимізований, виходячи зі складу робіт та існуючих вимог. Зміна вимог може привести до істотної зміни плану, а також дизайну проекту.

Адаптивні методології націлені на подолання очікуваної неповноти вимог і їх постійної зміни. Коли змінюються вимоги, команда розробників теж змінюється. Команді, що бере участь в розробці адаптивної методології, складно передбачити майбутнє проекту. Існує точний план лише на найближчий час. Більш віддалені в часі плани існують лише як декларації про цілі проекту, очікувані витрати і результати. Основне бажання замовника ПЗ – отримати готовий додаток високої

якості швидко і при мінімальних витратах на його розробку. Вкладаючи значні кошти на створення ПЗ, замовники часто бажають контролювати процес його розробки. Критерієм якості має бути задоволення вимог замовника на момент введення ПЗ в експлуатацію.

Розглянемо методології, яким практики відають перевагу у сучасному програмуванні.

Dynamic system development method (DSDM) – методологія, яка з'явилася в результаті роботи консорціум з 17-ти англійських компаній. Для DSDM розроблені посібники, навчальні курси, короткі серії семінарів, які передбачають: вивчення здійсненності програми та області її застосування; вивчення можливостей методології в межах обраного проекту; обговорення основних положень, що стосуються архітектури майбутньої системи та план проекту. Далі процес поділяється на три взаємопов'язані цикли: цикл функціональної моделі відповідає за створення аналітичної документації та прототипів; цикл проектування і конструювання – за приведення системи в робочий стан; цикл реалізації – забезпечує розгортання програмної системи.

SCRUM (з англ. «scrum» – штовханина, сутичка) – методологія, призначена для невеликих команд (до 10-ти осіб). Весь проект поділяється на ітерації (спринти) тривалістю 30 днів кожний. Обирається список функцій системи, які планується реалізувати протягом наступного спринту. Найважливіші умови SCRUM – незмінність певних функцій під час виконання однієї ітерації і суворе дотримання строків випуску чергового релізу, навіть, якщо до його випуску не вдасться реалізувати весь запланований функціонал.

KANBAN (японський термін, який почали використовувати у 60-х роках XX століття у компанії Toyota) – гнучка методологія розробки ПЗ, орієнтована на завдання. Основними правилами якої є візуалізація розробки: поділ роботи на завдання; використання позначок про стан завдання в розробці; обмеження робіт, що

виконуються одночасно, на кожному етапі розробки; вимірювання часу циклу (середній час на виконання одного завдання) та оптимізація процесу.

Зазначимо, що однією з умов забезпечення високої якості ПЗ є активне залучення користувачів до процесу розробки за методологією прототипного проектування. Ядром цієї методології є швидка розробка додатків Rapid Application Development (RAD).

RAD – концепція створення засобів розробки додатків, програмних продуктів, що приділяє особливу увагу швидкості й зручності програмування, створенню технологічного процесу, що дозволяє програмістові максимально швидко створювати комп'ютерні програми. Засновником RAD вважається співробітник IBM Джеймс Мартін, який в 1980-х роках сформулював основні принципи RAD. У даний час RAD стає загальноприйнятою схемою для створення засобів розробки ПП. Саме засоби розробки, засновані на RAD, мають найбільшу популярність серед програмістів. Дана технологія забезпечує на ранній стадії реалізацію прототипу. У науковій літературі відсутнє єдине розуміння поняття «прототипу». Однак, в межах нашого дослідження під цим поняттям ми розуміємо працюючу, первісну модель побудованої програми. Прототип дозволяє наочно продемонструвати користувачеві майбутню систему, уточнити її інтерфейсні елементи: форми введення повідомлень, меню, вихідні документи, структуру діалогу, склад функцій, що реалізовуються тощо. У процесі роботи з прототипом користувач реально усвідомлює можливості майбутньої системи і визначає найбільш зручний для нього режим обробки даних. Існує два базових варіанти використання прототипів: для кращої специфікації вимог до ПЗ, після розробки яких сам прототип стає непотрібним; для ітераційного розвитку прототипу в готовий ПП. Ітерації розробки прототипу включають створення/модифікацію прототипу, його демонстрацію користувачеві, узгодження, розробку нових вимог до ПЗ, нову модифікацію, поки не буде створено готовий додаток.

ЖЦ ПЗ при підході RAD має такі стадії:

а) аналіз і планування вимог: визначення функцій, які повинна виконувати система; виділення пріоритетних функцій, що вимагають опрацьовування насамперед; опис інформаційних потреб;

б) формулювання вимог, завдяки чому реалізуються наступні задачі: обмежується масштаб часу; встановлюються часові рамки для кожної з подальших стадій; визначається сама можливість реалізації проекту в заданих рамках фінансування та при наявних апаратних засобах;

в) проектування: користувачі, безпосередньо взаємодіючи з розробниками, уточнюють і доповнюють вимоги до системи, які не були виявлені на попередній стадії;

г) реалізація: безпосередньо виконується швидка розробка додатку, при якій розробники проводять ітеративну побудову реальної системи на основі отриманих на попередній стадії моделей, а користувачі оцінюють отримувані результати і вносять корективи;

д) виконання робіт: здійснюється аналіз використання даних і визначається необхідність їх розподілу; проводиться фізичне проектування бази даних (якщо вона має бути); формуються вимоги до апаратних ресурсів; встановлюються способи збільшення продуктивності; завершується розробка документації проекту;

е) впровадження: проводиться навчання користувачів, організаційні зміни і паралельно з впровадженням нової системи продовжується експлуатація існуючої системи.

Слід зазначити, що RAD крім переваг має ряд недоліків: застосовується для невеликих проектів, що розробляються для конкретного замовника; передбачає високий рівень планування і жорстку дисципліну проектування, строге слідування заздалегідь розробленим протоколам і інтерфейсам, що знижує швидкість розробки; не використовується для побудови складних розрахункових програм, систем

реального часу і операційних систем; 4) не застосовується для додатків, від яких залежить безпека людей.

Зазначимо, що паралельно із методологією RAD у другій половині 1990-х років широкого застосування набула методологія Rational Unified Process (RUP), розроблена в компанії Rational Software. Ця методологія описує абстрактний загальний процес, на основі якого організація або проектна команда повинні створити спеціалізований процес, орієнтований на її потреби. RUP передбачає використання об'єктно-орієнтованого моделювання.

Виділяють такі основні характеристики процесу RUP:

- Розробка вимог. Використовуються прецеденти або варіанти використання. Повний набір варіантів використання (ВВ) системи разом з логічними відношеннями між ними називається моделлю варіантів використання [2]. Кожний ВВ – це опис сценарію взаємодії користувача з системою, що повністю виконує конкретну задачу користувача. Згідно RUP всі функціональні вимоги мають бути представлені у вигляді ВВ. Вважається, що модель ВВ дає більш цілісне уявлення про функціональність системи в порівнянні з традиційним описом вимог (перерахуванням функцій, якими повинна володіти система).
- Ітеративна розробка. Проект в RUP складається з послідовності ітерацій з рекомендованою тривалістю від 2-х до 6-ти тижнів. Ітеративна модель дозволяє вносити необхідні зміни до вимог, проектних рішень і реалізації в ході проекту.

RUP – це методологія, орієнтована на виконувану архітектуру, що дозволяє реалізувати архітектурно значущі ВВ. Основи виконуваної архітектури мають бути реалізовані якомога раніше. Це дозволяє оцінити адекватність ухвалених архітектурних рішень і внести необхідні корективи ще на початку проекту. Згідно методології RUP, загальний ЖЦ системи складається з декількох циклів-ітерацій. Кожний цикл-ітерація розбивається на 4 фази або етапи:

- початок – ця фаза складається з однієї ітерації, в ході виконання якої необхідно визначити бачення і межі проекту; створити економічне обгрунтування;

ідентифікувати більшу частину ВВ і детально описати декілька ключових ВВ; знайти хоча б одне можливе архітектурне рішення; оцінити бюджет, графік і ризики проекту;

- проектування – на основі вимог і ризиків проекту створюється основа архітектури системи та цілями цієї фази є: детальний опис більшої частини ВВ; створення відтестованої базової архітектури; зниження основних ризиків і уточнення бюджету та графіка проекту;
- побудова – розробка остаточного продукту, створюється основна частина початкового коду системи і випускаються проміжні демонстраційні прототипи;
- впровадження – проведення бета-тестування і тренінгів користувачів, виправлення виявлених дефектів, розгортання системи на робочому майданчику, ухвалення рішення про досягнення мети.

У термінах RUP учасники проектної команди створюють так звані артефакти (Work Products), виконуючи завдання в рамках певних ролей.

Артефакти в RUP – це модель, елемент моделі, документ, вихідний код [3; 5].

У RUP визначені інженерні та допоміжні дисципліни: бізнес-моделювання; управління вимогами; аналіз і проектування; реалізація; тестування; розгортання; управління конфігураціями і змінами; управління проектом; середовище. Крім переваг RUP має ряд недоліків:

- часто послідовність фаз “початок – проектування – побудова – впровадження” помилково інтерпретують як фази “аналіз – дизайн – реалізація – впровадження”;
- незвична концепція документування вимог у вигляді ВВ – у результаті специфікація вимог, створена проектною командою, може виявитися невиразним документом, який ніхто не читатиме;
- для повноцінного впровадження RUP організація повинна витратити значні кошти на навчання співробітників.

Тому не випадково, що на практиці, починаючи з 1994 року, крім методології RAD, широкого впровадження набула методологія Microsoft Solutions Framework (MSF), яка спирається на практичний досвід корпорації Майкрософт і описує управління людьми і робочими процесами в ході розробки рішення. MSF складається з принципів, моделей і дисциплін по управлінню персоналом, процесами, технологічними елементами і пов'язаними зі всіма цими чинниками питаннями, характерними для більшості проектів. MSF включає в себе дві моделі і три дисципліни:

- модель проектної групи і модель процесів;
- дисципліни управління проектами, ризиками, підготовкою.

До проектної групи входять такі рольові кластери: управління програмою, розробка, тестування, управління випуском, навчання користувача, управління продуктом. Модель процесів представляє загальну методологію розробки і впровадження рішень інформаційних технологій (IT). Особливість цієї моделі полягає в тому, що, завдяки своїй гнучкості і відсутності жорстко регламентованих процедур, вона може бути застосована при розробці досить широкого кола IT-проектів. Вона поєднує в собі властивості двох стандартних виробничих моделей: каскадної і спіральної. Модель процесів в MSF була доповнена ще одним інноваційним аспектом: вона покриває весь ЖЦ створення рішення, починаючи з його відправної точки і закінчуючи безпосередньо впровадженням. Такий підхід допомагає проектним групам сфокусувати свою увагу на бізнес-віддачі рішення, оскільки ця віддача стає реальною лише після завершення впровадження і початку використання продукту. Модель процесів MSF враховує постійні зміни проектних вимог. Вона виходить з того, що розробка рішення повинна складатися з коротких циклів, які дозволяють пересуватися від простих версій рішення до його остаточного вигляду. Процес розробки за MSF є ітеративним і включає наступні основні фази: вироблення концепції, планування, розробка, стабілізація, впровадження. Рішення не представляє бізнес-цінності, поки воно не впроваджене. Саме з цієї причини

модель процесів MSF містить весь ЖЦ створення рішення, включаючи його впровадження, аж до моменту, коли рішення починає давати віддачу.

Управління ризиками – невід'ємна частина ЖЦ ІТ, що пропонує принципи, ідеї і рекомендації, підкріплені описом покрокового процесу для успішного активного управління ризиками. Цей процес включає: виявлення і аналіз ризиків; планування і реалізацію стратегій по їх профілактиці; відстеження стану ризиків.

Проект – обмежена часовими рамками діяльність, мета якої полягає у створенні унікального продукту або послуги. Добре відомою є взаємозалежність між ресурсами проекту, його календарним графіком і можливостями, що реалізовуються. Ці три змінні утворюють так званий «трикутник компромісів». Управління підготовкою – це також одна з ключових дисциплін MSF. Вона присвячена управлінню знаннями, професійними уміннями і здібностями, необхідними для планування, створення і супроводу успішних рішень. Дисципліна управління підготовкою описує фундаментальні принципи MSF і дає рекомендації по застосуванню превентивного підходу до управління знаннями впродовж всього ЖЦ ІТ. Ця дисципліна також розглядає планування процесу управління підготовкою.

Зазначимо, що крім переваг MSF має деякий недолік: можливість ефективного використання лише у командній розробці.

Розглянемо модель прототипування (рис.1.4), ключовими поняттями якої є: прототипування – це процес побудови робочої моделі системи; прототип – це еквівалент експериментальної моделі або «макета» у світі апаратного забезпечення. Використання даної моделі передбачає послідовне створення макетів системи, які будуть все ближче і ближче до реального продукту.

«Швидка» часткова реалізація системи відбувається перед етапом визначення вимог або протягом цього етапу. «Кінцеві» користувачі системи використовують прискорений прототип, а потім шляхом зворотного зв'язку повідомляють про своє досягнення команді, яка працює над проектом, для подальшого уточнення вимог до системи. Процес уточнення продовжується до того часу, доки користувач не отримає

те, що йому потрібно. Після завершення процесу визначення вимог шляхом розробки прискорених прототипів, отримують детальний проект системи, а прискорений прототип регулюється при використанні коду або зовнішніх утиліт, в результаті чого отримують кінцевий робочий продукт. В ідеалі можна створити, при чому без зайвих витрат, модель прототипування високої якості, не економлячи на документації, аналізі, проектуванні, тестування та ін.

Рисунок 1.4 – Модель прототипування

Такий підхід має перевагу, яка полягає в тому, що на кожному кроці ми маємо систему, яка працює. Хоча така система і не володіє всією потрібною функціональністю, але покращується із кожною ітерацією. При цьому, не витрачаються ресурси на код, який буде «викинуто».

Еволюційний підхід до прототипування може бути обраний, виходячи з припущення, що всі необхідні вимоги до моменту початку розробки невідомі, і будуть визначатися при створенні програми. Тоді на кожному етапі реалізують лише ті вимоги, які відомі і зрозумілі. Іноді при цьому розробники зосереджуються на роботі тільки над тими модулями системи, вимоги на які вже визначені.

Таким чином, приходимо до висновку, що для створення нашого ПП доцільно використовувати модель еволюційного прототипування з огляду на те, що вона має низку переваг у порівняні з іншими моделями: дозволяє виконувати гнучке проектування та розроблення, включаючи кілька ітерацій на всіх фазах ЖЦ; мінімізує можливість виникнення розбіжностей при спілкуванні замовників із розробниками; збільшує імовірність задоволення користувачів, які беруть участь у процесі розробки протягом усього ЖЦ.

Висновки до розділу 1

У даному розділі було розглянуто основні етапи атаки та основні теоретичні дані про атаки, з ціллю її подальшого представлення у вигляді моделі. Також було розглянуто теорію мереж Петрі, як інструмент для представлення моделей в роботі, та теорію графів вцілому. Приклад моделі здійснення атаки на інформаційну систему було представлено в даному розділі.

Виконана порівняльна характеристика сучасних методологій дозволила встановити, що для запланованого нами проекту доцільно використовувати ітеративну МЖЦПЗ з огляду на те, що дана модель включає в себе необхідні для передбачуваного нами ПП такі складові: можливість зміни вимог до кінцевого виду ПП, отримання по завершенню кожного з циклів розробки прототипу, для оцінки коректності поставленого перед нами завдання.

РОЗДІЛ 2. МЕТОДИ ДОСЛІДЖЕННЯ РІВНЯ КІБЕРБЕЗПЕКИ IT-ІНФРАСТРУКТУРИ

2.1 Тестування на проникнення

Дослідження інформаційної безпеки можливе з використанням двох способів. Перший – це аудит системи, а другий – це проведення тестів на проникнення (penetration test, pentest) [16].

Аудит – це методика оцінки на відповідність вимогам, кращим практикам або рекомендаціям нормативних актів, стандартів і документації виробників обладнання та програмного забезпечення такими як, Cobit, стандартами серії ISO / IEC 2700x, рекомендаціями CIS / SANS / NIST / etc і стандартом PCI DSS.

Тести на проникнення або пентести – метод оцінки безпеки комп'ютерних систем або мереж засобами моделювання атаки зловмисника.

Набір методів і засобів заснований на моделюванні атаки зловмисника. Включає в себе активний аналіз системи на наявність потенційних вразливостей і їх використання, аналіз ведеться з позиції потенційного атакуючого і включає в себе активне використання вразливостей системи. Об'єктами тестування можуть бути як окремі інформаційні системи, наприклад: CMS (система управління вмістом), CRM (система управління взаємовідносинами з клієнтами), інтернет клієнт-банк, так і вся інфраструктура в цілому: периметр мережі, бездротові мережі, внутрішня або корпоративна мережа, а так же зовнішній периметр.

Тестування на проникнення не може гарантувати безпеку для системи [6, 12], оскільки список потенційних небезпек повністю невідомий і, отже, незліченний.

Тестування на проникнення може забезпечити лише обмежену перспективу безпеки поточної системи [13]. Ефективність тестування проникнення багато в чому залежить від майстерності та досвіду тестерів. Щоб отримати хороші достовірні результати тесту на проникнення, тестер і його методи і знання повинні бути дуже

глибокими. Незважаючи на ці вимоги, тести на проникнення, які слідують методології, можуть стати більш ефективними у використанні ресурсів і забезпечити більш достовірні результати [14].

Використання певної методології є корисним при проведенні тестування на проникнення. Тестування на проникнення залежить від багатьох механізмів для виявлення недоліків в атаках або тестах, тому структурований підхід, таким чином, може принести користь як тестерам, так і ресурсам, що використовуються під час тесту.

Додатковою перевагою є те, що результати тесту з хорошою структурою легше повторно використовувати в майбутнього, щоб забезпечити відсутність регресії [15]. У літературі про пентести, більшість методологій подібні [1, 12, 14]. Кожен крок у методології має різні назви, але кожен з подібними значеннями. Нижче описані деякі з найбільш використовуваних методологій для тестування проникнення.

Більшість підходів подібні до того, як зловмисник атакує систему. Основні етапи підготовки:

– виявлення;
– перерахування;
– відображення вразливості;
– експлуатація.

Це типова методологія, представлена для тестування на проникнення в літературі з безпеки [12]. Подібні книги мають лише невеликі відмінності, наприклад, як поділ 4-го кроку на отримання доступу і привілейовану експлуатацію.

2.2 Методи розвідки

Під час проведення тестів на проникнення використовується великий набір сервісів та утиліт для визначення сигнатур важливих про визначення вразливостей, таких як: ОС та її версія, сервіси та їх версії та інше [18]. У LAN мережах нелегко з'ясувати, які сервіси і мережі належать конкретної організації. Однак в Інтернеті

існує безліч спеціальних інструментів, що дозволяють без особливих зусиль визначити мережі, підконтрольні цікавлять нас компаніям, і при цьому ніяк не засвітитися перед ними. Для пасивної розвідки в рамках збору статистики по мережевим периметрах фінансових організацій хакери зазвичай використовують:

– пошукові системи (Google, Shodan);

– галузеві сайти для фінансового сектора - Rbc.com;

– whois-сервіси 2ip.com; nic.com;

– пошукові системи по базах даних - Hurricane Electric BGP Toolkit, RIPE;

– сервіси візуалізації даних по доменному імені сайту – Robtex, зображення результату пошуку сервісу представлений на рисунку 2.1.

– сервіс для аналізу доменних зон dnsdumpster, який містить в собі історичні дані за доменною зоною (зміни IP-адреси), чим сильно допомагає зібрати дані. Схожих сервісів багато, один з найвідоміших аналогів - domaintools.com.

Після визначення мережі, які належить цільовій організації, визначаються IP-адреси за допомогою сервісу whois. Він видає наступну інформацію:

– ім'я (мережеве ім’я, дуже корисно при пошуку по БД RIPE - надає можливість вільного пошуку всіх зареєстрованих в компанії підмереж);

– descr (опис може бути застосовано для пошуку з використанням фантазії);

– адреса (пошук сетей, зареєстрованих на цьому ж фізичному адресі);

– контакт (можливий пошук в БД RIPE по людям, які також можуть зареєстровані в мережі);

– інша інформація, за якою можна ідентифікувати організацію.

Для ідентифікації доступних послуг можна використовувати один із двох найвідоміших інструментів, призначених зробити безпечнішими комп'ютерні мережі: Shodan або Censys [19] [20].

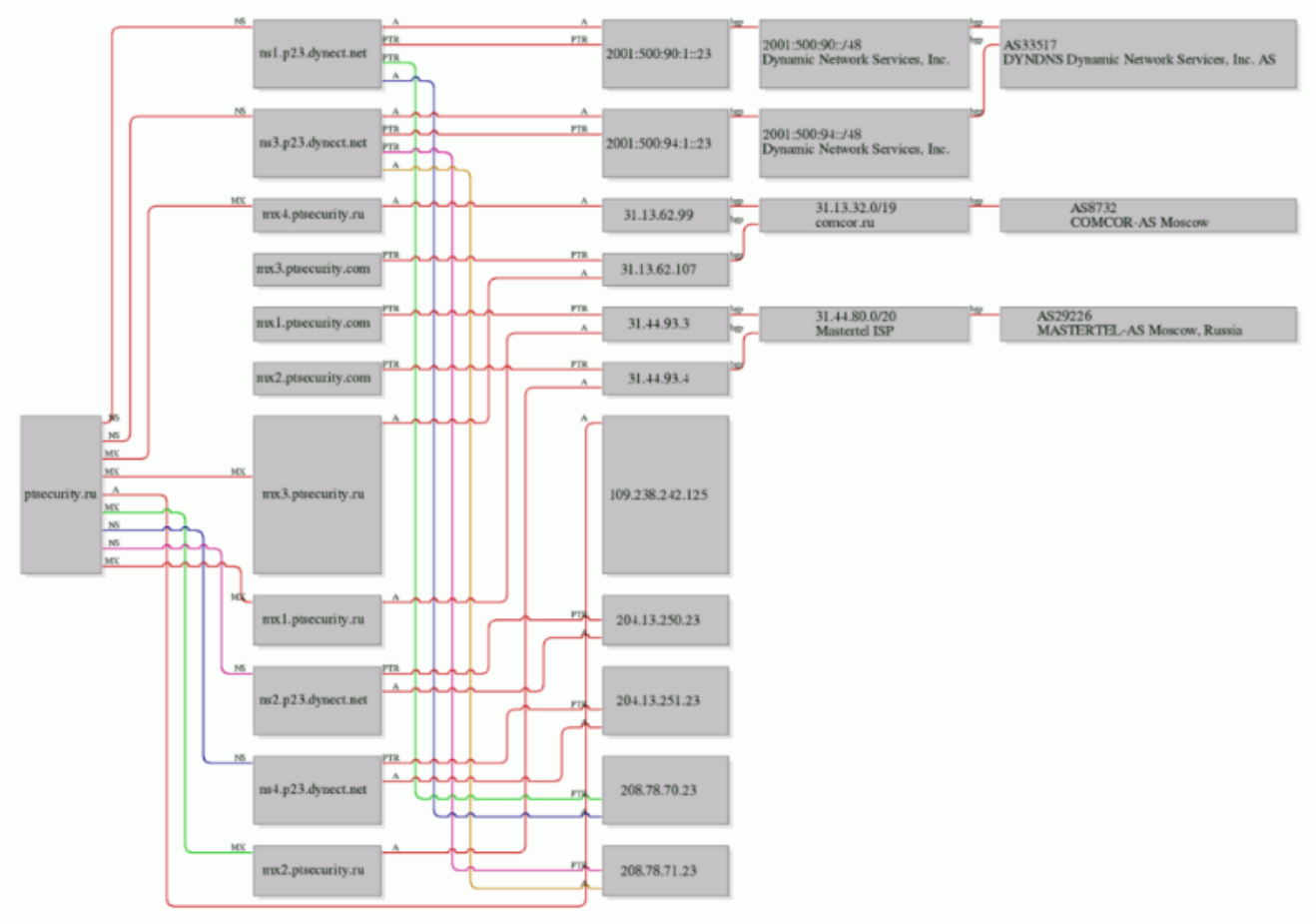


Рисунок 2.1 - Результату пошуку під-доменів кореневого домену за допомогою сервісу Robtex.

Вони мають подібні можливості, підтримують роботу з API. Для повного пошуку обом службам потрібна реєстрація, щоб отримати ключ API. На рисунку 2.2 представлено результат сканування певного хосту використовуючи доступ через браузер. Інформація, яку можна отримати після сканування Shodan:

– IP – унікальна мережева адреса вузла в комп'ютерній мережі, побудована за допомогою протоколу IP;

– порт – числове число, яке є параметром транспортних протоколів (таких як TCP і UDP);

– протокол – набір конвенцій інтерфейсу логічного рівня, що визначають комунікацію між різними програмами;

– ім'я хоста – символічне ім'я, присвоєне мережевому пристрою, яке можна використовувати для організації доступу до цього пристрою різними способами;

– сервіс – назва конкретного сервісу;

– продукт – назва програмного забезпечення, з яким реалізується послуга;

– версія продукту – конкретна версія програмного забезпечення;

– банер – вітальна інформація, надана службою при спробі підключитися до неї;

– перелік загальних платформ CPE, стандартизований спосіб іменування програмних програм, операційних систем та апаратних платформ;

– ОС – версія операційної системи.

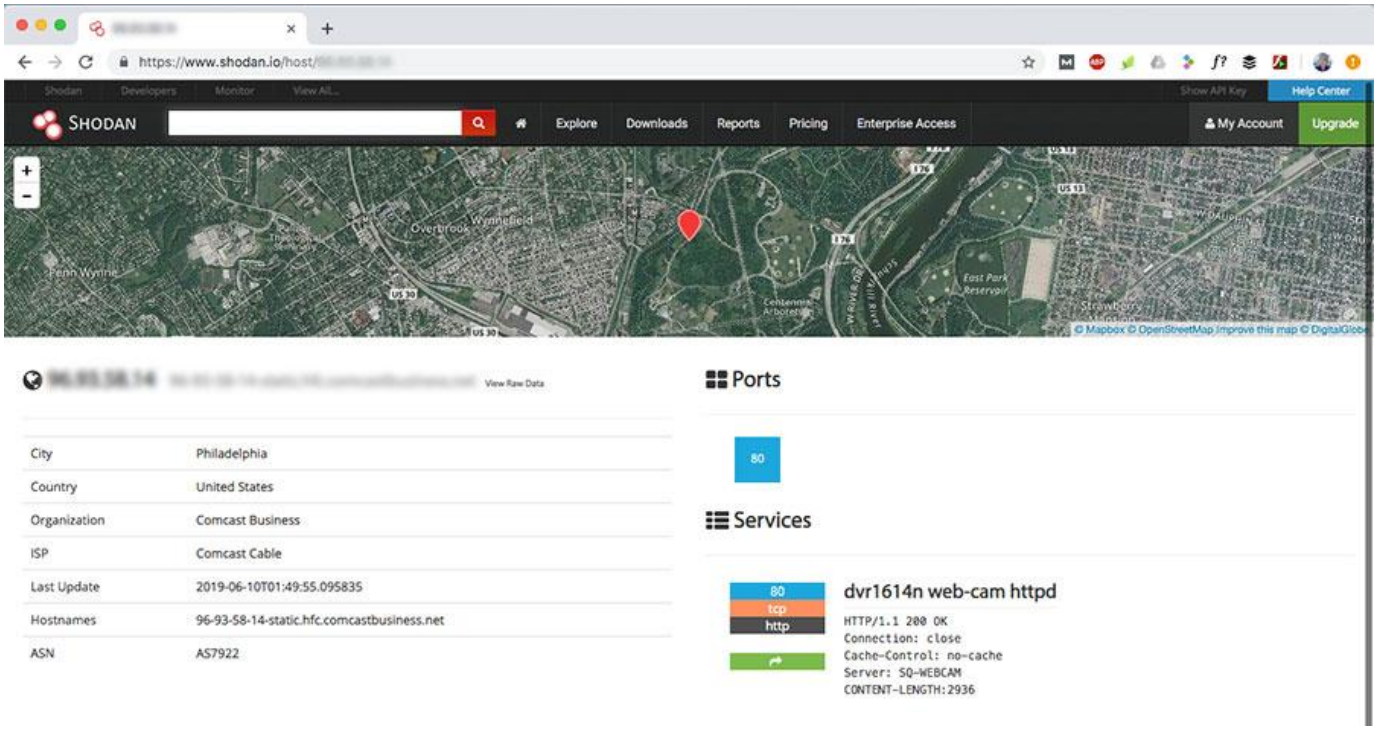


Рисунок 2.2 - Результат сканування сервісом Shodan

IP-адреса, порт і CPE – це дані, якими користуються програми. IP та порт дуже важливі, щоб знати про місце розташування потенційної вразливості в цільовій інформаційній системі. CPE – найкращий спосіб визначити наявність будь-якої

вразливості в Національній базі даних вразливостей і в позитивному випадку знайти ідентифікаційний номер цієї вразливості [21].

2.3 Інструменти та експлойти

Щоб допомогти процесу тестування проникнення, необхідні інструменти та код експлуатації. Повністю ручне тестування в сучасних комплексних інформаційних системах неможливе в межах ресурсів, які зазвичай використовуються для тестування проникнення.

Книги написані про засоби та методи, які використовуються при тестуванні на проникнення, пояснюють використання багатьох інструментів і пропонують інструментарій для пентестів.

Обговорюються також помилки деяких категорій інструментів. Прикладами є засоби відмови в обслуговуванні, які можуть мати небажані наслідки для цільових та інших інструментів, які мають помилки, які можуть ненавмисно завдати шкоди цілі.

У мережі існує чимала кількість загальнодоступних джерел для пошуку вразливостей, ось деякі з них:

– БДУ ФСТЕК - база даних загроз інформаційній безпеці, що відрізняється від інших подібних ресурсів можливістю знайти уразливості для ПО вітчизняного виробництва;

– nvd.nist.gov - національна база даних вразливостей Інституту стандартів і технологій США, яка об'єднує загальнодоступні державні ресурси США з пошуку та аналізу вразливостей;

– vulners.com - велика база даних ІБ-контенту, дозволяє шукати уразливості, експлойти, патчі, результати неправильної роботи коду;

– cvedetails.com - зручний веб-інтерфейс для перегляду даних про уразливість. Ви можете переглядати список постачальників, продуктів, версій і CVE (Common Vulnerabilities and Exposures) пов'язаних з ними вразливостей;

– securityfocus.com - один з найвідоміших загальнодоступних джерел, особливо в частині наповнення даними по експлойтів.

– National Vulnerability Database (NVD) є урядовим сховищем урядових стандартів даних управління вразливістю на основі стандартів, представлених за допомогою протоколу автоматизації контенту безпеки (SCAP). Ці дані дозволяють автоматизувати управління вразливістю, вимірювання безпеки та відповідність. NVD включає бази даних посилань на контрольний список безпеки, недоліки програмного забезпечення, пов’язані з безпекою, неправильні конфігурації, назви виробів та показники впливу.

NVD забезпечує якісні рейтинги загрози, які зображені в таблиці 2.1, “Low”(низький), “Medium”(середній) та “High”(високий) для базових шкал CVSS v2.0 на додаток до оцінок суворості для CVSS v3.0, як вони визначені у специфікації CVSS v3.0 “Critical”(критичний) та “None”(відсутній).

Таблиця 2.1 - Якісні рейтинги загрози вразливості сервісу за шкалами CVSS v2.0 та CVSS v3.0

CVSS v2.0 Ratings		**CVSS v3.0 Ratings**	
Загроза	**Базовий діапазон балів**	**Загроза**	**Базовий діапазон балів**
		Відсутня	0.0
Низька	0.0-3.9	Низька	0.1-3.9
Середня	4.0-6.9	Середня	4.0-6.9
Висока	7.0-10.0	Висока	7.0-8.9
		Критична	9.0-10.0

Загальна система оцінювання вразливості (CVSS) – це відкритий фреймворк для передачі характеристик та рівня вразливості програмного забезпечення. CVSS складається з трьох метричних груп: базової, часової та середовища. Базові показники дають бал від 0 до 10, який потім можна змінити, оцінюючи часові та

екологічні показники. Оцінка CVSS також представлена у вигляді векторного рядка, стислого текстуального подання значень, використовуваних для отримання оцінки. Таким чином, CVSS цілком підходить як стандартна система вимірювань для галузей, організацій та урядів, які потребують точних та послідовних показників на скільки виражена вразливість. Два загальні методи використання CVSS - це обчислення ступеня вираженості вразливих ситуацій, виявлених в системах, і як фактор пріоритетності діяльності з усунення вразливості. Національна база даних про вразливість (NVD) надає бали CVSS майже для всіх відомих вразливостей.

NVD підтримує як загальну систему оцінювання вразливості (CVSS) v2.0, так і v3.X. NVD забезпечує базові показники CVSS, які представляють вроджені характеристики кожної вразливості. Наразі NVD не надає "тимчасових показників" (показники, які змінюються в часі через події, що не стосуються вразливості) або "екологічні показники" (бали, налаштовані так, щоб відображати вплив вразливості на вашу організацію). Однак NVD постачає калькулятор CVSS як для CVSS v2, так і для v3, що дозволяє додавати дані про тимчасову та екологічну бали. CVSS належить та керується FIRST.Org, Inc. (FIRST), некомерційною організацією, що базується в США, місією якої є допомога командам реагування на комп’ютерну безпеку в усьому світі.

2.4. Ідентифікація вразливостей

Бази даних та реєстри вразливостей корисні широкому колу фахівців в області інформаційної безпеки. Мережеві адміністратори або співробітники, відповідальні за безпеку комп'ютерних систем організації, можуть своєчасно дізнатися з баз та реєстрів про появу нових загроз для захищаються ними систем, визначити пріоритетні заходи реагування на ці загрози (виходячи з оцінки критичності і поширеності в організації уразливого ПЗ).

При цьому для фахівців важливі оперативність відновлення баз даних вразливостей, зручність отримання цих оновлень і ступінь покриття обраним реєстром вразливостей як основних видів ПЗ захищається комп'ютерної системи, так і всієї множини виявляються для цього ПЗ вразливостей. Також значимими характеристиками будуть наявність рекомендацій щодо усунення вразливостей і можливість визначення потенційних векторів атак на захищаються комп'ютерні системи.

Створення реєстрів і баз даних з інформацією про виявлені вразливості зажадало уніфікованого (принаймні в рамках однієї бази даних) способу їх ідентифікації та класифікації. Кожній виявленої уразливості потрібно присвоїти ідентифікатор, дати їй місткий і короткий опис, визначити для неї список уразливого ПО і його версій. Крім того, потрібно оцінити ступінь критичності даної уразливості по ряду різних критеріїв:

– простоті виявлення зловмисником уразливого ПО;
– легкості експлуатації уразливості;
– необхідним для цього привілеїв;
– потенційно можливих наслідків експлуатації уразливості для комп'ютерної системи.

Ця інформація могла згодом доповнюватися рекомендаціями щодо усунення вразливості або запобігання її експлуатації і статусом уразливості (чи присутня вона в актуальній версії ПЗ або була закрита відповідними оновленнями і патчами).

Дані про індивідуальні вразливості в сукупності з організованим зберіганням, функціями пошуку по базі і автоматичного отримання повідомлень про нові виявлені вразливості можуть бути корисні широкому колу фахівців в області інформаційної безпеки - ми більш детально повернемося до цього питання в ув'язненні статті.

Кінець 90-х і початок 2000-х років характеризувалися появою цілого ряду каталогів і баз даних вразливостей, підтримкою яких займалися самі різні, незалежні

один від одного, організації. Серед них можна можна виділити кілька основних типів:

– офіційні дослідні організації і групи реагування на інциденти, створювані на основі різних урядових установ і наукових закладів;

– некомерційні організації, що підтримують контакти як з незалежними дослідниками з комп'ютерної безпеки, так і виробниками ПО для координації зусиль з пошуку нових вразливостей;

– некомерційні організації, що координують діяльність окремих незалежних дослідників в області інформаційної безпеки і мають власний штат фахівців-ентузіастів;

– комерційні організації, що спеціалізуються на наданні послуг в області інформаційної безпеки і захисту інформації; для даних організацій підтримка власного реєстру вразливостей ПЗ - не тільки важливий інструмент для власної діяльності, а й публічна демонстрація компетенції фахівців компанії.

Поява і розвиток незалежних один від одного баз даних і реєстрів вразливостей породило відому проблему синхронізації інформації між ними, як в частині однозначної ідентифікації вразливостей (посилаються ці два записи в різних джерелах на одну і ту ж або на різні вразливості), так і в класифікації і оцінці вразливостей, викликаних відмінностями в системах, що надають оцінки.

Стандарт Common Vulnerabilities and Exposures (CVE), розроблений американською некомерційною дослідницької корпорацією MITRE Corporation в 1999 році, де-факто є на сьогоднішній день основним стандартом в області уніфікованого іменування і реєстрації виявлених вразливостей програмного забезпечення. Даний стандарт безпосередньо визначає як формат ідентифікаторів і вмісту записів про окремих виявлених вразливостей, так і процес резервування ідентифікаторів для нових виявлених вразливостей і поповнення відповідних баз даних.

Уже 2000 року ініціатива MITRE зі створення єдиного стандарту для реєстрації та ідентифікації виявлених вразливостей ПЗ отримала широку підтримку з боку провідних виробників програмного забезпечення і дослідницьких організацій в області інформаційної безпеки. В даний час підтримкою і адмініструванням реєстру вразливостей CVE займається група з 84 організацій по всьому світу, в число яких входять провідні виробники програмного забезпечення, телекомунікаційного обладнання і інтернет-сервісів, такі як Apple, Cisco, Facebook, Google, IBM, Intel, Microsoft, Oracle та ряд компаній, що сеціалізуються в області інформаційної безпеки, наприклад, F5 Networks, McAfee, Symantec, «Лабораторія Касперського» і ін.

Це створило додаткові проблеми як для адміністраторів самих реєстрів вразливостей, так і незручність для їх користувачів, і в кінцевому підсумку призвело до ініціативи MITRE CVE.

В рамках підтримки проекту MITRE CVE основними завданнями цих організацій (званих CVE Numbering Authorities, CNAs), є:

– пошук і збір інформації про уразливість програмного забезпечення (в разі виробників і вендорів ПО, їх область відповідальності обмежена безпосередньо їх власними продуктами і сервісами);

– класифікація знайдених вразливостей;

– резервування CVE-ідентифікаторів для знайдених вразливостей;

– актуалізація відповідної інформації в двох офіційних каталогах - реєстрі вразливостей CVE List самої MITRE Corporation і базі даних вразливостей NVD, підтримуваної національним Інститутом Технологій та Стандартів США.

На поточний день бази вразливостей MITRE CVE List і NVD містять близько 98 тисяч записів про окремих вразливості, виявлених за період з 1999 року по теперішній час. При цьому, хоча самі бази даних розрізняються на рівні функціональних можливостей, що надаються користувачам, самі списки записів про уразливість фактично ідентичні один одному. Формально CVE List виступає

початковим джерелом записів для бази даних NVD, а фахівці, що відповідають за підтримку бази NVD, виробляють уточнений аналіз і збір доступної інформації вразливостей, зареєстрованим в CVE List (наприклад, збирають посилання на сторонні джерела інформації про уразливість і заходи щодо її усунення або запобігання експлуатації).

Ідентифікатори CVE мають формат CVE-YYYY-NNNN, відображаючи в перших чотирьох цифрах рік реєстрації уразливості і в наступних чотирьох-шести цифрах - унікальний в рамках цього року номер уразливості.

Для кожної з виявлених вразливостей запис в базі містить короткий опис типу і причин уразливості, вразливі версії ПЗ, оцінку критичності уразливості відповідно до стандарту CVSS (Common Vulnerability Scoring System) і посилання на зовнішні джерела з інформацією про уразливість - найчастіше, такими виступають інформаційні бюлетені на сайтах виробників програмного забезпечення або дослідницьких організацій.

У плані призначеного для користувача функціоналу в CVE List підтримуються можливості найпростішого пошуку серед записів (за ключовими словами і CVE-ідентифікаторів) і скачування архівів записів за будь-який обраний рік в різних форматах (HTML, XML, CVRF, CSV або Plain Text). Також можливе автоматичне отримання оновлень в машиночитаемом вигляді через спеціальний data feed CVE Change Log (він дозволяє як відстежувати появу нових ідентифікаторів CVE, так і зміни в записах для вже існуючих).

Для бази NVD в свою чергу доступні просунуті функції пошуку вразливостей за ключовими словами, тимчасовими діапазонами створення \ модифікації запису, компонентам CVSS-метрики і т. П. Крім того, доступні скачування всіх записів бази даних в XML, а також отримання інформації про оновлення бази у вигляді RSS-підписки і JSON data feed.

Перевагою баз даних MITRE CVE List і NVD є щоденне оновлення реєстрів відомих вразливостей. При виявленні нової уразливості виробником ПО або

дослідницькою організацією (або підтвердження наявності уразливості вендором ПО у відповідь на повідомлення від приватних дослідників або організацій, що не входять в CVE Numbering Authorities) під неї оперативно реєструється новий ідентифікатор CVE і створюється запис в базі, після чого відбувається періодичне оновлення інформації.

В середньому, за добу в базах CVE List і NVD реєструється 13 нових записів про виявлені вразливості. При цьому унікальність реєстрованого ідентифікатора забезпечується ієрархічною структурою CNAs (як показано на малюнку), в якій кореневі організації (Root CNA) ділять і розподіляють між підлеглими організаціями (Sub CNA) діапазон доступних в цьому році ідентифікаторів CVE. Кожна з відповідних підрядних організацій в свою чергу розпоряджається наданими діапазоном ідентифікаторів для створення записів про виявлені вразливості в своїх власних продуктах, або виявлених вразливостей в продуктах третьої сторони, за умови, що вона не є учасником CVE Numbering Authorities.

Слід зазначити, що в даний час серед учасників CVE Numbering Authorities лише дві організації мають статус кореневих CNAs, що знаходяться під безпосереднім адмініструванням Primary CNA (самої MITRE Corporation). На рисунку 2.3 зображена ієрархія структури CVE Numbering Authorities.

Сильною стороною самого стандарту CVE є його повсюдна підтримка в сучасних програмних продуктах і сервісах, спрямованих на забезпечення інформаційної безпеки. Далеко не повний перелік видів цих продуктів і сервісів включає: бази даних і реєстри вразливостей (власні записи в цих базах містять в якості зовнішніх посилань і CVE-ідентифікатори вразливостей, якщо для уразливості взагалі був привласнений CVE-ідентифікатор), системи виявлення \ запобігання атак, антивірусні засоби (CVE-ідентифікатори в сигнатурних правилах), сканери безпеки і засоби моніторингу та ін.

Деяким природним обмеженням баз даних CVE List і NVD є відсутність в записах про уразливість будь-якої інформації про точне місце локалізації уразливості

в коді уразливого ПО і можливих вектори атак, за допомогою яких можлива експлуатація даної уразливості. У деяких випадках ця інформація може бути знайдена по посиланнях на зовнішні ресурси, проте в більшості випадків виробники і вендори ПО уникають публікації даної інформації, причому не тільки на період розробки та впровадження патчів, що закривають виявлену уразливість, а й в подальшому. Частково така політика пояснюється небажанням учасників CVE Numbering Authorities надавати подібну інформацію потенційним зловмисникам, особливо в світлі того, що вразливе програмне забезпечення може бути широко поширене по всьому світу, а експлуатують його організації часто не мають можливостей або не надають належного значення своєчасній установці оновлень.

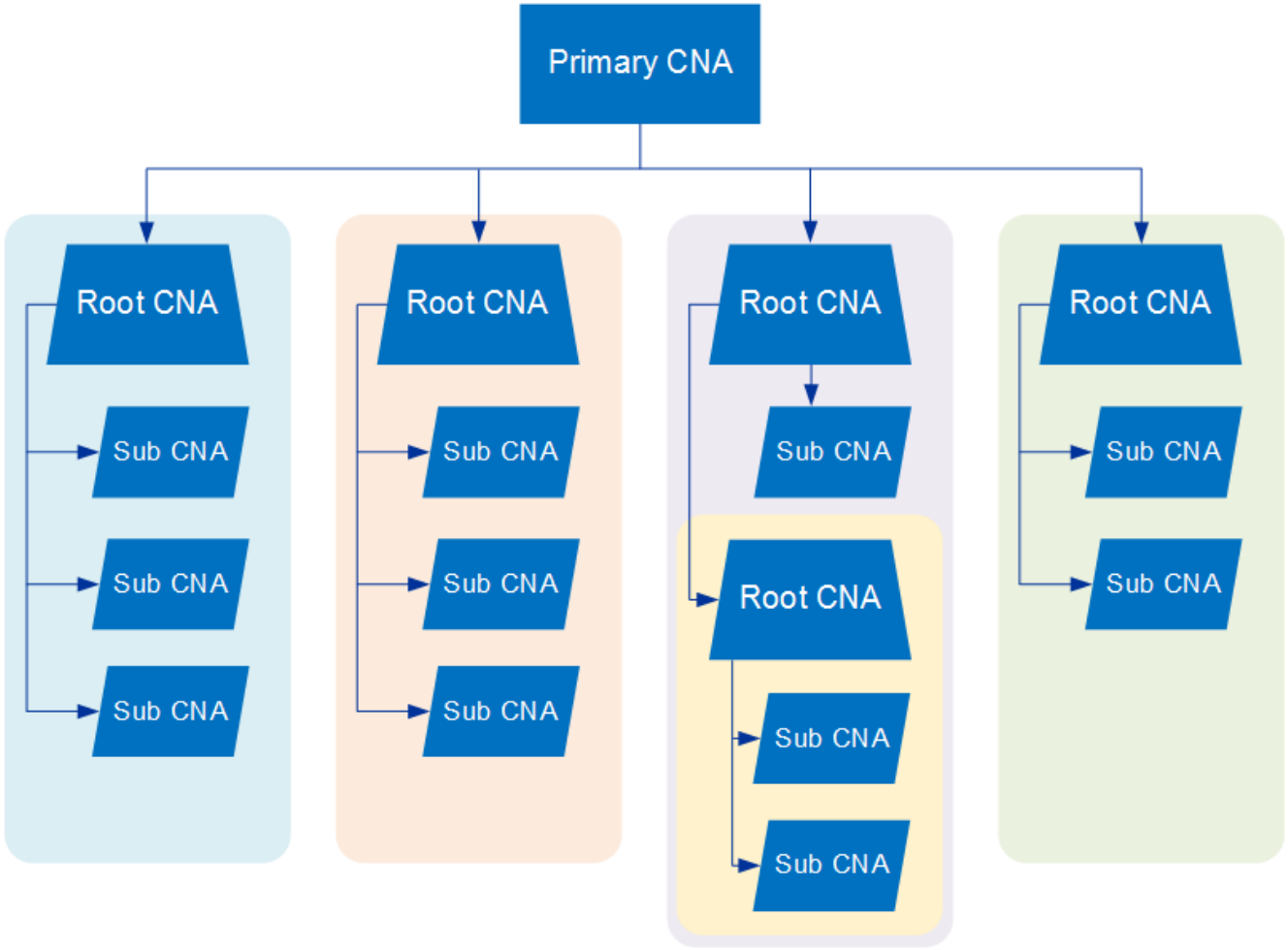


Рисунок 2.3 - Ієрархічна структура CNAs

2.5 Агрегатори інформації про вразливість

Різноманітність різних реєстрів і баз даних вразливостей (їх загальне число в кілька разів більше, ніж було розглянуто в статті) викликає у фахівців в області інформаційної безпеки (в першу чергу, розробників засобів захисту, фахівців з тестування на проникнення і дослідників, які шукають і тих, хто вивчає нові уразливості ПО) природне бажання використовувати різного роду агрегатори інформації, які б забезпечували автоматизований збір доступної інформації про уразливість і додаткові функції пошуку і філ ції, що цікавить.

Подібного роду агрегатори інформації про уразливість існують і представлені різного роду сервісами, починаючи від спеціалізованого агрегатора CVE-релевантної інформації і до агрегатора з інтерфейсом повноцінної пошукової машини, адаптованої під предметну область.

Прикладом сервісів першого типу є CVEDetails – простий спеціалізований агрегатор інформації про уразливість, який збирає всю доступну з публічних реєстрів і баз даних вразливостей інформацію по конкретному CVE-ідентифікатором і об'єднує її в єдину запис.

Фактичним функціоналом даного сервісу є автоматизація пошуку всієї доступної інформації по CVE-ідентифікатором з додатковими функціями пошуку по вендорам, типам вразливостей, оцінці критичності по метриках CVSS і т. П. Також реалізовані збір і зберігання різного роду статистики вразливостей, наприклад, розподіл вразливостей за ступеня критичності (згідно метриці CVSS), розподіл вразливостей за вендорам ПЗ та ін. - з зручним переходах до списку вразливостей, які відповідають обраним критерієм.

Що стосується інтерфейсу, то CVEDetails в цілому орієнтований на компактне і зручне для сприйняття людиною табличне представлення даних, а для автоматизованих систем підтримує формування RSS-підписки (в форматі JSON) для

отримання оновлених даних про уразливість обраних категорій, наприклад, для всіх нових вразливостей класу SQL-ін'єкцій або XSS.

Прикладом іншого підходу є Vulners – розроблений фахівцями і вельми популярний серед експертів в області інформаційної безпеки сервіс з власною базою даних, призначений для пошуку інформації з найрізноманітніших матеріалів в області інформаційної безпеки (включаючи публікації на тематичних ресурсах, бюлетені вендорів, інформацію про заходи Bug Hunting і фахівцях, безпосередньо які виявили вразливості та ін).

Фактично Vulners є пошуковий движок з власною базою даних, адаптований під предметну область. Таким чином він покриває набагато ширшу множину сутностей, ніж прості агрегатори вразливостей.

В даний час база даних Vulners агрегувала в себе близько 870 тисяч записів про уразливість і приблизно 170 тисяч записів про відомих експлойтів. За даним масиву інформації можливі пошук за ключовими словами і фільтрація результатів як за джерелом інформації (організації, що опублікувала запис про уразливість), так і за датою публікації записи, CVSS-оцінці критичності уразливості і іншим подібним параметрами.

Слід зазначити, що Vulners не надає якоїсь єдиної зведення інформації по конкретній уразливості з заданим CVE-ідентифікатором (або іншим внутрішнім ідентифікатором одного з альтернативних реєстрів), а повертає безліч записів, релевантних пошуковому запиту в стилі класичного пошукового движка. При цьому наявність фільтрації результатів по організації-джерела інформації (наприклад, type: cvelist) дозволяє робити вибірку записів тільки з вказаною бази даних.

Всі результати пошукової видачі з бази даних Vulners можуть бути отримані не тільки в зручному для людини, але і в вигляді зручному для розробників вигляді (у форматі JSON) через відповідний API пошукового запиту

Висновки до розділу 2

В даному розділі розглянуто мінуси та плюси тестування на проникнення та різноманітний інструментарії для ідентифікації вразливостей. Розробники засобів і систем захисту комп'ютерних мереж (сканерів безпеки, систем виявлення і запобігання атак, міжмережевих екранів, антивірусних засобів і ін.) Отримують завдяки моніторингу різних баз даних вразливостей цінну інформацію про нові можливі атаки на комп'ютерні системи. Ця інформація включає супровідні дані, корисні для формування звітів адміністратору системи захисту. На основі цієї інформації можна визначити ступінь критичності атаки, що важливо і в момент прийняття рішення про реагування, і на стадії формування звіту. У деяких випадках користувачам доступна і готова інформація про вектор атаки, що дозволяє припустити як буде виглядати атакуючий трафік, який сервіс буде метою атаки та інше.

РОЗДІЛ 3. АРХІТЕКТУРА ЗАСТОСУВАННЯ

3.1 Запропонований метод моделювання інформаційних атак

Моделювання атак здійснювалося в спеціально розробленому середовищі об'єктно-орієнтованою мовою програмування Java. Компоненти мережі Петрі (місця, переходи, дуги, лексеми) представлені у вигляді об'єктів, і всі процеси відбуваються методами ООР. У додатку використовується графічна оболонка для спрощення створення моделі. Особливістю програми є перетворення мережі Петрі в метод Java після візуалізації моделі, що надає можливість повторного використання малих деталей моделі для створення великої моделі [22].

На рисунку 3.1 показаний фрагмент мережі Петрі для моделювання проникнення хакерів. Така взаємодія відбувається лише тоді, коли хост-сервіс має доступну вразливість, яку в ідеальному випадку може використовувати хакер. Поодинокі вузли (CVE-2019-10097, CVE-2019-10092, CVE-2019-10082 та CVE-2019-10098) представлені уразливими на рисунку 3.1.

Загальні вразливості та експозиції (CVE) – це перелік вразливостей інформаційних систем – кожна з яких містить ідентифікаційний номер, опис і принаймні одне публічне посилання на публічно відому кібернетичну вразливість, його формат: CVE-0000-0000 [23] [24] [25]. Всі представлені CVE є вразливістю сервісу Apache HTTP Server версії 2.4.39. Зазвичай його назву спрощують до Apache. Apache був обраний через його популярність; веб-сервер є першим у топ-5 веб-серверів [26].

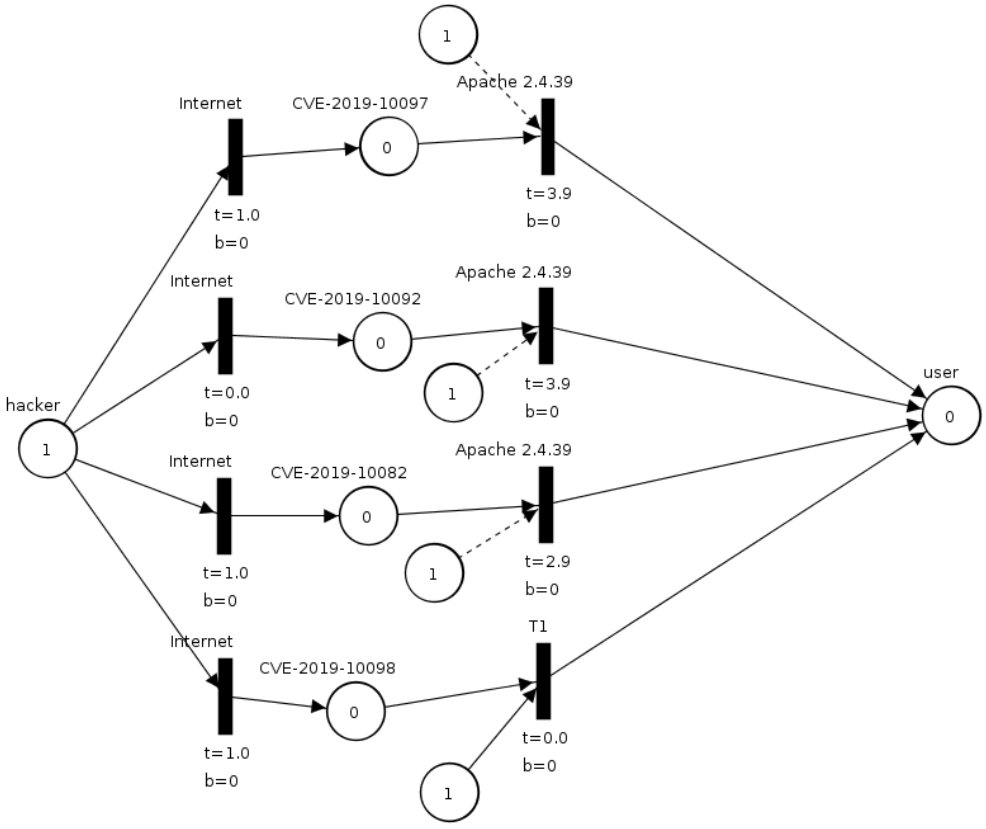


Рисунок 3.1 - Фрагмент мережі Петрі, що моделює проникнення хакеру

В застосуванні було використано сервіс Shodan для визначення сервісів, які є відкритими на хості, точніше CPE сервісу. Shodan має REST API, також Github представив його реалізацію для різних мов програмування, що є особливо важливо, для Java [27] [28]. Для автоматизації процесу передачі всіх CPE до бази NVD використовувся фреймворк Selenium для Java [29]. Selenium - це основа для управління браузерами за допомогою драйвера WebDriver для різних браузерів. Додаток використовує WebDriver для сбору та аналізу інформації на сайтах, як було зазначено вище, це сайт Національної бази даних про вразливості (NVD). Після отримання відомих CVE, застосування використовуючи Selenium збирає на сайті джерела даних CVE інформацію про вразливості, які потім використовуються як вхідні данні для моделювання за допомогою мережі Петрі.

CVE Details – це ресурс, який допомагає виявити основні характеристики вразливості [30]:

– тип вразливості – тип вразливості, який визначається CWE (Common Weakness Enumeration) – список розроблених спільнотою типів слабкості програмного забезпечення [31];

– автентифікація – тип правил, яким хакер повинен користуватися вразливістю (root або користувач);

– отриманий рівень доступу – правило, яке хакер може отримати після використання вразливості (root або користувач);

– доступ – тип доступу (мережевий, локальний);

– складність – метрика описує, наскільки легко чи важко використовувати виявлену вразливість [20].

Всі ці характеристики визначаються загальною системою оцінювання вразливості (CVSS – Common Vulnerability Scoring System), яка є вільним і відкритим стандартом для оцінки ступеня вираженості потенційних або фактичних вразливих ситуацій в комп'ютерних системах [32].

У моделі існує прямий взаємозв'язок між складністю та затримкою в часі: чим нижчий показник складності, тим довше хакер буде його використовувати. Складність вразливості оцінюється відносно її небезпеки, тобто чим складніше її експлуатувати, тим менш небезпечно, адже не кожен хакер може мати достатньо навичок. У таблиці 3.1 представлений повний опис метрики складності зі значеннями та балами [33].

Таблиця 3.1 - Повний опис метрики складності зі значеннями та балами

Значення	**Опис**	**Оцінка**
Висока (H - high)	Існують спеціалізовані умови, такі як стан перегонів із вузьким вікном або вимога до методів соціальної інженерії, які легко помітять обізнані люди.	0.35

Продовження таблиці 3.1

Середня (M - medium)	Існують деякі додаткові вимоги до атаки, такі як обмеження походження атаки або вимога для роботи вразливої системи з нечастою конфігурацією, що не використовується за замовчуванням.	0.61
Низька (L - low)	Не існує спеціальних умов для використання вразливості, наприклад, коли система доступна великій кількості користувачів або вразлива конфігурація є усюдисущою.	0.71

3.2 Опис архітектури програмного застосування

Програмне забезпечення розроблено об’єктно орієнтованою мовою програмування – Java. Складається з двох модулів:

– модуль, розроблений за рахунок модифікації бібліотеки для візуалізації моделей з використанням Петрі об’єктів, в якій графічний редактор реалізовано за допомогою стандартної бібліотеки JavaFX [22];

– модуль, який збирає і віддає вхідні данні бібліотеці, для побудови мережі Петрі.

Для запуску програмного забезпечення необхідно в якості аргументу передати ім'я, IP-адресу або пул адрес. Ці дані передаються для обробки в методі, що відповідає за Shodan сканування. Сервіс використовує Shodan REST API, який повертає JSON файл на REST запит, але для нього створено зручні бібліотеки, що спрощують виконання запиту і створені для багатьох мов програмування, зокрема для Java – jShodan [38]. Shodan збирає сигнатури вразливостей і записує їх в базу даних PostgreSQL. Застосування включає в себе ORM Hibernate [37] та Spring Data [39] для спрощення розробки ПЗ та обробки запитів до бази даних.

Адміністратор безпеки має заповнити базу, додавши інформацію про внутрішні вузли системи, до яких немає зовнішнього доступу, але має доступ відкритий хост. Окрім IP адрес треба додати дані про встановлені сервіси.

Наступним кроком вступає в дію сервіс, що відповідає за збір даних з бази NVD, тобто за збір CVE ідентифікаторів вразливостей. NVD не має зручного API, є лише можливість завантажити базу в JSON форматі, але по-перше, там міститься не повна інформація, по-друге, таку базу потрібно постійно скачувати, оскільки вона оновлюється щодня. Через перераховані вище причини, було вирішено робити прямі запити на сайт, що імітують дії користувача за допомогою фреймворку Selenium [33] для Java. Selenium має в своїй структурі WebDriver – драйвер, що імітує роботу браузера. Результатом роботи даного програмного сервісу є заповнена база даних, у відповідність до кожного сервісу ставиться набір вразливостей, якщо вони були знайдені.

Для побудови логіки моделі наступним кроком є збір даних про кожну вразливість, а саме:

– вектор враження(Vector Attack) – локальний або віддалений;
– складність експлуатації вразливості (Access Complexity);
– права для експлуатації вразливості (Authentication);
– права, які отримує зловмисник після експлуатаціїї вразливості (Gained Access).

Тепер в роботу вступає перший програмний модуль, він забирає зібрану першим модулем інформацію та генерує метод в класі NetLib, даний метод можна запустити в графічній оболонці. Модель почне непередбачувано імітувати шляхи проходу хакером до цілі. Результатом є список всіх пройдених шляхів та час їх проходження. Модель дасть більш точний результат, тобто всі можливі шляхи, якщо поставити більший час її виконання.

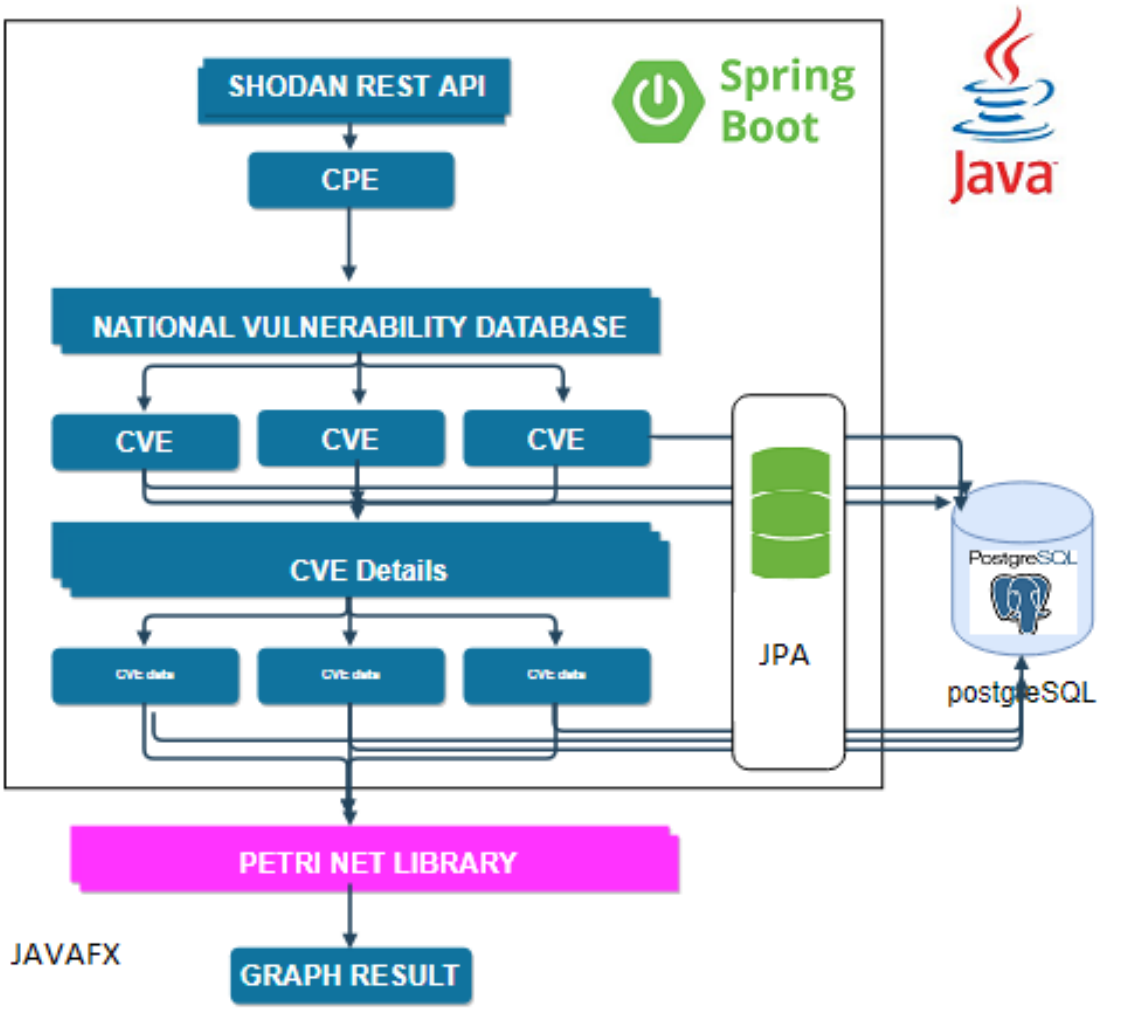


Рисунок 3.2 - Архітектура розробленого застосування.

Висновки до розділу 3

В даному розділі описано модель, що імітує розповсюдження атаки зловмисником, вказано, які вхідні дані використовуються для побудови мережі Петрі. Також описано архітектуру програмного застосування, яка включає в себе сервіси для визначення сигнатур вразливостей та самих вразливостей з деталями про них. Ці дані перед побудовою мережі зберігаються в базі даних PostgreSQL [39], а сама модель у вигляді методу, тому шляхом визову методу з графічного редактору.

РОЗДІЛ 4. РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ

Моделювання здійснювалося за типовою архітектурою, яка складається з веб-сервера, файлового сервера та бази даних. З Інтернету можна потрапити лише до тієї мережі, в якій розміщений веб-сервер та файловий сервер. На рисунку 4.1 показана архітектура експериментальних мереж. Мережа, в якій розміщений веб-сервер та файл, має прямий доступ до Інтернету. У таблиці 4.1 наведені дані про вразливості хостів, зібрані на попередніх етапах програми. На рисунку 4.2 показана мережа Петрі поширення атаки на дані вразливості сервісів відповідних IP.

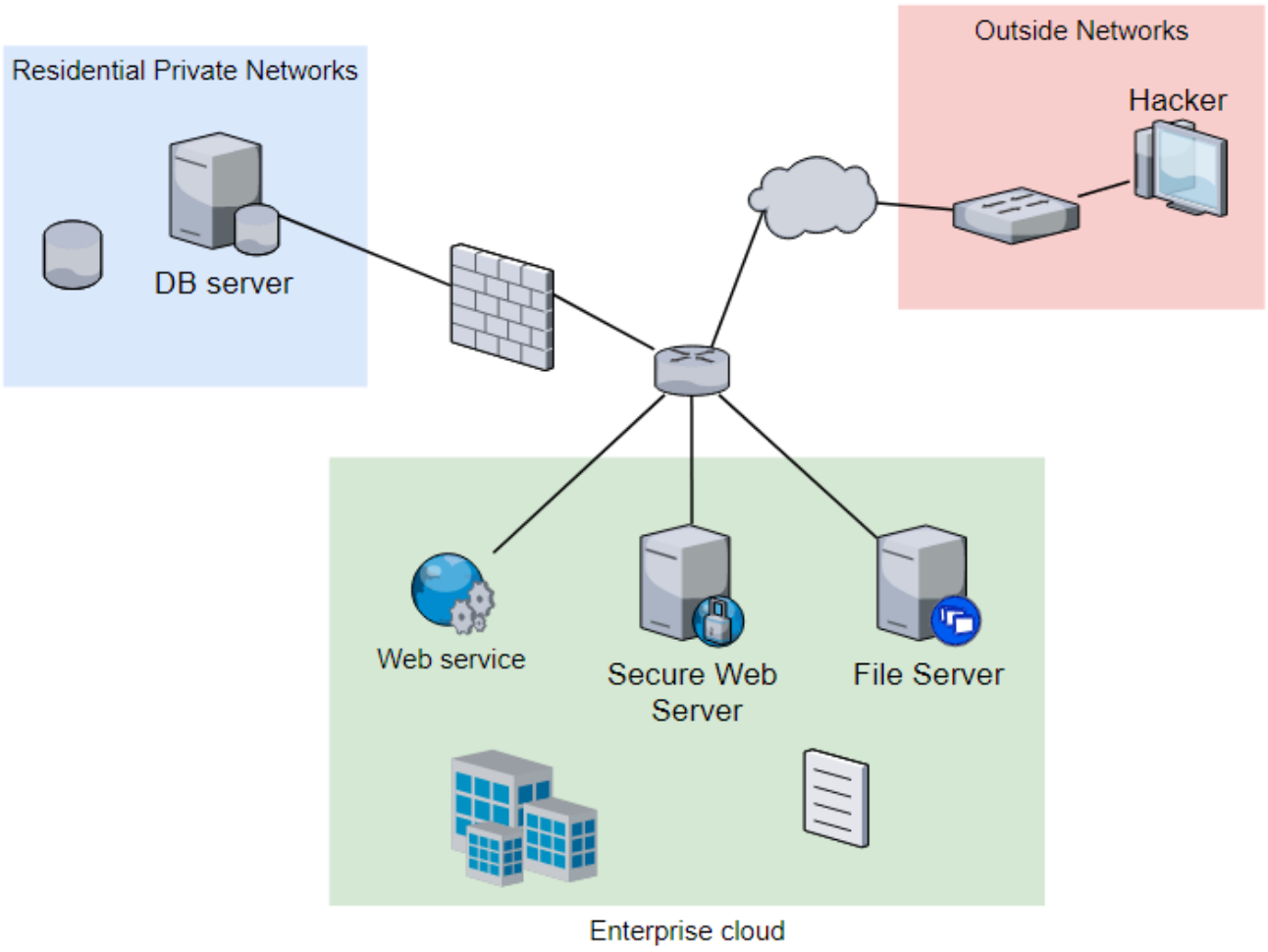


Рисунок 4.1 - Архітектура інформаційної системи, що моделюється

Таблиця 4.1. Дані про вразливості хостів

ID хостів	**CVE ID**	**Тип доступу**	**Складність**	**Аутентифікація**	**Набутий доступ**
172.31.172.31/24	CVE-2019-10097	віддалений	середня	single system	ніякий
	CVE-2019-10092	віддалений	середня	not required	ніякий
	CVE-2019-10082	віддалений	низька	not required	ніякий
	CVE-2019-10098	віддалений	середня	not required	ніякий
	CVE-2016-0727	локальний	низька	not required	адмін
172.31.172.32/24	CVE-2018-1057	віддалений	низька	single system	ніякий
	CVE-2018-1050	віддалений	середня	not required	ніякий
	CVE-2011-2411	віддалений	низька	single system	ніякий

Продовження таблиці 4.1

10.31.10.31/24	CVE-2007-3278	віддалений	середня	not required	адмін
	CVE-2016-5424	віддалений	висока	single system	ніякий
	CVE-2016-0727	локальний	низька	not required	адмін

Модель Петрі будується на основі даних представлених в попередній таблиці (додаток А).

Очікування в переходах, що моделюють експлуатацію вразливості в сервісі $\left(1-(score_{complexity})\right)\cdot 10$, оскільки 10 секунд є найпоширенішим порядком затримок у часі.

Результатами моделювання є час проходження від початку атаки до її закінчення певним шляхом:

Path 1(CVE-2019-10097, CVE-2016-0727, CVE-2016-5424, CVE-2016-0727) 17.2s

Path 2(CVE-2019-10092, CVE-2016-0727, CVE-2016-5424, CVE-2016-0727) 17.2s

Path 3(CVE-2019-10082, CVE-2016-0727, CVE-2016-5424, CVE-2016-0727) 16.2s

Path 4(CVE-2019-10098, CVE-2016-0727, CVE-2016-5424, CVE-2016-0727) 17.2s

Path 5(CVE-2018-1057, CVE-2016-0727, CVE-2016-5424, CVE-2016-0727) 16.2s

Path 6(CVE-2018-1050, CVE-2016-0727, CVE-2016-5424, CVE-2016-0727) 17.2s

Path 7(CVE-2011-2411, CVE-2016-0727, CVE-2016-5424, CVE-2016-0727) 16.2s

Path 8(CVE-2019-10097, CVE-2007-3278) 11.4s

Path 9(CVE-2019-10092, CVE-2007-3278) 11.4s

Path 10(CVE-2019-10082, CVE-2007-3278) 10.4s

Path 11(CVE-2019-10098, CVE-2007-3278) 11.4s

Path 12(CVE-2018-1057, CVE-2007-3278) 10.4s

Path 13(CVE-2018-1050, CVE-2007-3278) 11.4s

Path 14(CVE-2011-2411, CVE-2007-3278) 10.4s

З результатів легко зрозуміти, що вразливість CVE-2007-3278 є найбільш небезпечною; це скорочує час атаки на третину часу порівняно з іншими шляхами атаки. Найкращий спосіб у цьому випадку позбутися - змінити версію бази даних на нову. Остання версія PostgreSQL - 12.1, випущена 2019-11-14 [26]. На рисунку 7 показано, як змінити поширення атак в мережі після зміни версії служби бази даних.

Оновлена модель Петрі представлена в додатку А.

Отримані результати після заміни сервісу бази даних і, відповідно, моделі:

Path 1(CVE-2019-10097, CVE-2016-0727, CVE-2009-2943, CVE-2016-0727) 14.6s

Path 2(CVE-2019-10092, CVE-2016-0727, CVE-2009-2943, CVE-2016-0727) 14.6s

Path 3(CVE-2019-10082, CVE-2016-0727, CVE-2016-5424, CVE-2016-0727) 13.6s

Path 4(CVE-2019-10098, CVE-2016-0727, CVE-2009-2943, CVE-2016-0727) 14.6s

Path 5(CVE-2018-1057, CVE-2016-0727, CVE-2016-5424, CVE-2016-0727) 13.6s

Path 6(CVE-2018-1050, CVE-2016-0727, CVE-2016-5424, CVE-2016-0727) 14.6s

Path 7(CVE-2011-2411, CVE-2016-0727, CVE-2016-5424, CVE-2016-0727) 13.6s

З результатів можна побачити, що мінімальний час атаки зменишився на 30%.

Висновки до розділу 4

Основний результат, який було отримано, полягає в тому, що компанія, що моделює проникнення та розповсюдження атак у своїй інформаційній системі, може оцінити та знищити уразливість для подальшого їх виключення. Відповідно до результатів, що випливають після моделювання тестової архітектури інформаційної мережі, можна зробити висновок, що вразливості CVE-2019-10082, CVE-2018-1057 та CVE-2011-2411 слід відхилити в першу чергу. тому що хакер найшвидшим чином проникає в систему і досягає саме того, щоб отримати повний контроль над базою даних. Більше того, найнебезпечніша вразливість у системі - CVE-2007-3278 у базі даних, що скорочує навпіл шлях хакера до цілі. Основними параметрами, які вплинули на результати, є права доступу, необхідні для вразливості, і які можна отримати за його допомогою.

РОЗДІЛ 5. СТАРТАП

5.1 Опис ідеї проекту (товару, послуги, технології)

Організація системи захисту комп’ютерної мережі – є важкою задачею, при вирішенні якої необхідно брати до уваги велику кількість параметрів. Вплив цих параметрів часто протилежний, невизначений та непередбачуваний. Це пояснюється тим, що необхідно організовувати захист не мережі як такої, а мережі з усіма її функціонуючими системами, які вміщують велику кількість компонентів. Звичайною є така ситуація, коли на різних мережевих вузлах встановлено різні операційні системи (іноді навіть дві або декілька одночасно), жорсткі диски мають різні файлові системи, склад категорій користувачів і їх права на використання ресурсів комп’ютера різні, та програмне забезпечення є дуже різноманітним. Оскільки всі ці факти впливають на можливість організації конкретних атак, побудова системи захисту – дуже трудомістка задача.

Використання комп’ютерних мереж зростає, саме тому кібербезпеці приділяється більше уваги, щодня адміністратори безпеки використовують інструменти для попередження підозрілої мережевої активності. У деяких ситуаціях доводиться мати справу з мільйонами таких попереджень протягом дня. Саме тому, необхідно ефективно перевіряти та ситуаційно оцінювати стан захищеності системи.

Оскільки при ефективному скануванні системи на вразливості може бути порушена робота сервісів та вузлів, тому альтернативою в даному випадку може стати створення моделі даної системи для проведення тестів на проникнення і зменшити витрати на проведення дорогих пентестів.

Мережі Петрі є одним з широко використовуваних формалізмів при моделюванні атак в комп'ютерних мережах. У термінах теорії множин мережу Петрі можна визначити як четвірку <P, T, I, O >, де $P=\{p_1, p_2,..., p_n\}$ —закінчена множина

місць ,n >= 0;T={t_1, t_2,..., t_m} –закінчена кількість переходів, m>=0, причому множини місць і переходів не перетинаються (рис.5.1).

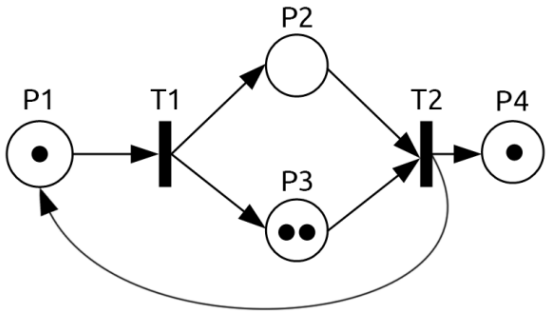


Рисунок 5.1 - Представлення мережі Петрі

Маркування мережі Петрі є функція, що відображає безліч місць P в безліч N, яке залежить від класу мережі Петрі. Маркування мережі Петрі призначена для моделювання її динамічних характеристик. Множина n може бути комплектом, тобто закінченою сукупністю об'єктів, в якій можлива присутність однакових об'єктів. Вихідна функція мережі Петрі може містити пост-операцію D , тобто операцію зміни структури мережі Петрі шляхом вставки або видалення місць або переходів.

Через можливість динамічного моделювання, враховувати багато параметрів та простоту використання було обрано використання Петрі-об’єктної моделі у створенні програмного додатку для аналізу проникності системи.

Було розглянуто готове програмне забезпечення для моделювання, в якому можна побудувати модель на основі мереж Петрі, таке як Arena Simulations. Це програмне забезпечення містить готові сегменти для моделей, але у випадку моделювання інформаційної системи необхідно будувати обширніші сегменти, що в сукупності створює громіздку та незручну у використанні систему. З цих причин було вирішено було створити власну програму, щоб, по-перше, створити фреймворк для мереж Петрі, а потім його імплементувати саме до моделі інформаційної системи підприємтва.

Запрограмувати таку модель було вирішено за допомогою високорівневої об’єктно-орієнтованої мови Java. Одним з основних переваг мови Java є можливість перенесення програм з однієї системи в іншу. Оскільки програми на Java не залежать від платформи як на рівні вихідного коду, так і на довічним рівні, їх можна запускати в різних системах, що особливо важливо для програм, призначених для World Wide Web.

Також для зручного користування моделлю клієнтами, буде використано графічну бібліотеку JavaFX.

Вирішенням такої ситуації може бути система, що спроможна моделювати процес атакуючих впливів в залежності від наявності або відсутності всіх вище вказаних (а також інших) факторів.

Перші три пункти подаються у вигляді таблиці (табл. 5.2) і дають цілісне уявлення про зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів.

Таблиця 5.1- Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Моделювання (на основі мереж Петрі) проникнення в інформаційну систему з використанням хакерських атак в залежності від вразливостей системи.	1.Моделювання	1. Оцінка ризиків (співставлення необхідності забезпечення певного рівня безпеки відносно цінності інформації, що циркулює в інформаційній системі)
	2.Захист	2. Заощадження коштів (понижується необхідність залучення експертів для проведення тесту на проникнення)

Продовження таблиці 5.1

Це дозволить організаціям перевіряти власну систему на вразливості змінюючи певні конфігурації власної системи та оцінювати ризик проникнення відносно архітектури системи та використаних сервісів для захисту	3. Оцінка ризиків	3. Отримання інформації про час поширення атаки у разі проникнення хакера до системи (в залежності від побудови архітектури мережі, наявності критичних модулів у системі)
		4. Отримання інформації про можливість реагування на атаку або її попередження
		5. Оцінка часу відновлення системи у разі хакерської атаки

Аналізуючи таблицю 5.1, можна зробити висновок, що потенційними клієнтами будуть компанії, які прагнуть захистити власну систему та інформацію, що в ній циркулює, та заощадити кошти і не наймати фахівців з проведення тестів на проникнення.

Аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників) порівняно із пропозиціями конкурентів передбачає:

– визначення переліку техніко-економічних властивостей та характеристик ідеї (орієнтований можливий перелік властивостей та характеристик подано у додатку А);

– визначення попереднього кола конкурентів (проектів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та проводиться збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;

– проводиться порівняльний аналіз показників: для власної ідеї визначаються показники, що мають а) гірші значення (W, слабкі); б) аналогічні (N, нейтральні) значення; в) кращі значення (S, сильні) (табл. 5.2).

Таблиця 5.2 - Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів			W (слабка сторона)	S (сильна сторона)
		проект	CPN(Colored Petri Net) tools	Arena simulations		
1.	Економічні	Для створення проекту використовуватиметься open source середовище для розробки на мові програмування Java, що знижує його вартість.	Безкоштовне програмне забезпечення для створення мереж Петрі	Безкоштовним використання даного ПЗ є лише у пробний період (30 діб).	Мій проект не буде безкоштовним	Поєднання ціна і спектр послуг роблять мій проект найвигіднішим серед існуючих концепцій.

Продовження таблиці 5.2

2.	Призначення (технічні)	Використання мереж Петрі в їх початковому вигляді, самостійне моделювання їх на Java, що в результаті буде готовим продуктом для моделювання атак на мережі.	Моделювання мереж відбувається за допомогою мереж Петрі, але в них втрачена основна властивість мереж – це простота. Використовуються різні додані харакатеристики, які ускладнюють моделювання і розуміння моделі.	Лідер у програмному забезпеченню для моделювання та автоматизації дискретних подій, але для створення найпростішої моделі атаки необхідно створити надзвичайно складну мережу.	Не можливо передбачити створення майбутніх атак для їх моделюванню	Проект немає аналогів у світі

Продовження таблиці 5.2

3.	Технологічні	Java Virtual Machine дозволяє запускати свій код на усіх відомих ОС.	Також існують версії для Linux та Windows.	Версія лише для Windows		Навіть враховуючи відсутність прямих аналогів у світі, проект буде максимально універсальним і легким для користувачів.

Аналізуючи таблицю 5.2 можна зробити висновок, що проект потенційно є конкурентноспроможнім, адже техніко-економічні характеристики ідеї є відкритими та доступними у користуванні, не потребують коштів та є відносно у своєму викоританні.

5.2 Технологічний аудит ідеї проекту

Технологічний аудит ідеї проекту представлений в таблиці 5.3.

Таблиця 5.3 - Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
	Змоделювати хакерську атаку	Технологія для створення моделей, яка може бути застосована для створення моделі атаки	Arena Simulations — використовується для моделювання, має вже запрограмований інтерфейс. Дана технологія наявна, але має бути дороблена через те, що одиниці моделі дуже прості і для створення одиниці запланованої моделі необхідно створити ще одну модель.	Технологія на даний момент недоступна, автори, по-перше, мають недостатню кваліфікацію в користуванні даною програмою та ,по-друге, необхідно її купити.
		Табличне або матричне представлення інформації	Недоступне при моделюванні великої кількості зав’язків між можливими інцидентами або відповідними діями порушника	Технологія доступна

Продовження таблиці 5.3

		Логічні моделі	Можливо використовувати при нечіткій логіці, що є плюсом у випадку невизначеності вихідних даних про атаку. Для моделювання логічного виводу зазвичай використовується віртуальна машина, що потребує значної потужності, необхідної для моделювання атак у великих комп'ютерних мережах.	Технологія доступна
		Графові моделі	Добре використовується коли потрібне масштабування, пов'язане в даному випадку для мереж з великою кількістю вузлів та вразливосте	Технологія доступна
		Баєсові графи	Можливо асоціювати вершини графа з інцидентами (елементарними умовами), а ребра – кон'юкція і диз'юнкцію умов. Також можна задати імовірності для кожної вершини, з допомогою яких можна розрахувати імовірність настання інциденту, використавши формулу умовної імовірності.	Технологія доступна

Продовження таблиці 5.3

		Мережі Петрі	Можливо динамічно зображати характеристики. Не дивлячись на свою простоту можуть зображати різноманітні варіації	Технологія доступна
		Кольорові мережі Петрі	Ускладнені мережі за рахунок ведення маркування, надають такі самі можливості, як і звичайні мережі Петрі, але з розширеним функціоналом	Технологія доступна
		Java і Framework JavaFx	Мова програмування Java – найрозповсюдженіша мова високорівневого програмування	Технологія доступна, з якою добре ознайомлені всі учасники, що створюють проект.

Обрана технологія реалізації ідеї проекту: Обраною технологією стали мережі Петрі у зв'язці з мовою програмування Java. Причиною такого висновку стали відсутність програмного забезпечення, яке варте було б дорозробки, тому створення власного фреймворку для мереж Петрі на Java, з допомогою якого та JavaFX для візуалізації буде створено модель хакерських атак на інформаційну систему.

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Розрахунок економічної ефективності інноваційного проекту за методикою ЮНІДО

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації

проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Таблиця 5.4 - Вихідні дані для визначення доходності проекту

N п/п	Показники	Одиниці виміру	Значення
1.	Витрати на обладнання	$	30000
2.	Монтаж, навчання персоналу	$	0
3.	Термін роботи обладнання після вводу	років	5
4.	Річний обсяг замовлень	шт.	300
5.	Валютний депозит	%	15,0
6.	Фактор ризику	%	3,0
7.	Інфляція на валютному ринку	%	3,0

Поточні витрати по роках наведені в наступній таблиці 5.5.

Таблиця 5.5 - Поточні витрати на здійснення проекту

Витрати	1 рік	2 рік	3 рік	Всього
1. Зарплата	100 000	160 000	180 000	440 000
2.Нарахування	10 000	10 000	10 000	30 000
3. Інші	5 000	5 000	5 000	15 000
Всього	115 000	175 000	195 000	485 000

Визначення обсягу одноразових (капітальних) витрат по роках, доларах. :

Витрати на придбання обладнання здійснюються в 0-й рік (з моменту придбання обладнання починається відрахунок терміну виконання проекту);

0-й рік - -30 000

Вартість проведення тесту на проникнення в середньому 3000$, починається від 2000$ (https://hub.hacken.io/blog/how-much-does-penetration-test-cost-or-price-of-your-security), розроблене нами програмне забезпечення має коштувати менше, щоб бути рентабельними на ринку, тому зазначена ціна продукту — 1500$.

Визначення обсягу грошових потоків (чистого доходу + амортизації, враховуючи оподаткування прибутку за ставкою 22%), доларах:

1й рік = -115 000 (продукція не вироблялась);

2й рік = 300 000-175 000=125 000;

3й рік = 300 000-195 000=105 000;

Визначення норми дисконтування проекту (d):

d= a + b + c,

де a - ціна капіталу, a=0,15;

b - рівень ризику, b=0,03;

c - рівень ризику роботи на валютному ринку (інфляція), c=0,03.

d=0,15+0,03+0,03=0,21.

Визначення чистого дисконтованого доходу(ЧДД) та чистої поточної вартості (ЧПВ)

$$ЧДД = \sum_{t=0}^{n} \frac{(D_t - K_t)}{(1+d)^t};$$

де D_t - чисті доходи t-го періоду;

K_t - витрати t-го періоду.

Таблиця 5.6 - Розрахунок чистого дисконтованого доходу проекту

Роки	D	K	$\frac{1}{(1+d)^t}$	$\frac{D}{(1+d)^t}$	$\frac{K}{(1+d)^t}$	ЧДД	ЧПВ
0	0	30 000	1,00	0	30 000	-30 000	-30 000
1	0	115 000	0,83	0	95 450	-95 450	-125 000

Продовження таблиці 5.6

2	300 000	175 000	0,68	204 000	119 000	85 000	-40 000
3	300 000	195 000	0,56	168 000	109 200	58 800	18 800
Всього:	600 000	515 000	X	372 000	228 200	18 800	x

Термін окупності проекту ($Т_{ок}$) визначається на підставі попередніх розрахунків ЧДД та ЧПВ (табл. Д3):

$$T_{ок} = p + ЧПВ_p / ЧДД_{p+1},$$

де p – останній рік, коли ЧПВ = 0;

$ЧПВ_p$ – значення ЧПВ в р-му році (без мінусу);

$ЧДД_{p+1}$ - значення ЧДД в (р+1)-му році.

$$T_{ок} = 2 + 40\,000/58\,800 = 2{,}68 \text{ (року)}.$$

Термін окупності проекту дорівнює близько двох з половиною років.

Індекс доходності та середньорічна рентабельність проекту:

Індекс доходності (ІД) - це відношення сумарного дисконтованого доходу до сумарних дисконтованих витрат.

$$ІД = \sum_{t=0}^{n} \frac{D_t}{(1+d)^t} / \sum_{t=0}^{n} \frac{K_i}{(1+d)^t} = \frac{372\,000}{288\,800} = 1{,}6259;$$

Повинно витримуватись співвідношення ІД>1. Оскільки 1,6259>1, то по цьому показнику проект можна рекомендувати до впровадження.

Тоді середньорічна рентабельність проекту (R) буде:

$$R = ІД / n \times 100\% = 162{,}59 / 3 = 54.2\%.$$

Таким чином, даний проект є високорентабельним. Оскільки в перший рік його реалізації завершувались наукові дослідження та освоювалось обладнання та навчався персонал роботі на даному обладнанні, то грошові потоки в перший рік забезпечувались у вигляді вливання сторонніх інвестицій. Реалізація перших замовлень розпочинається тільки з другого року . За два з половиною роки компанія зможе розрахуватись з інвестором, а наступні роки буде працювати на власний

прибуток. Термін окупності даного проекту визначається періодом між другим та третім роками (коли ЧПВ > 0, то проект окупається).

Спочатку проводиться аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (табл. 5.7).

Таблиця 5.7 - Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	2
2	Загальний обсяг продаж, грн/ум.од	1 000 000/10
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Не виявлено
5	Специфічні вимоги до стандартизації та сертифікації	Оскільки програмний продукт стосується безпеки, то він має бути просертивікований по міжнародним стандартам ISO 9001\ ISO 14001 які підтверждуютьйого функціональінсть, надійність та ефективність.
6	Середня норма рентабельності в галузі (або по ринку), %	54,2%

Даний ринок є дуже привабливим для входження адже проблема безпеки компаній наразі є надзвичайно актуальною, але багато компаній не готові витрачати великі кошти на реалізацію повномасштабної системи безпеки та постійно її підтримувати з цих причин програмний продукт, який є відносно недорогим буде привабливим з точки зору попиту на ринку.

Надалі визначаються потенційні групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи (табл. 5.8).

Таблиця 5.8 - Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	У сучасному світі багато важливої та коштовної інформації знаходиться в електронному обігу, які необхідно захищати використовуючи засоби захисту інформаційної безпеки	Об’єкти інформаційної діяльності у яких циркулює державна, комерційна інформація або персональні дані.	Відмінності потенційних клієнтів не мають впливати на програмний продукт, адже цілю його створення є охоплення систем будь-якої складності.	до продукції: має повноцінно моделювати системи, оцінювати спроможність відображати хакерські атаки та показувати слабкі сторони системи - до компанії-постачальника: постійно підтримувати модель (обновлювати інформацію щодо появ нових загроз в інформаційному прсторі) та змінювати модель відносно змін самої інформаційної системи

Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл.5.9-5.10). Фактори в таблиці подавати в порядку зменшення значущості.

Таблиця 5.9 - Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Поява конкурентів	Стороння компанія зробить продукт схожий за функціоналом	Оцінить новий продукт, врахує його недоліки та зробить на них основний акцент при випуску свого власного продукту
2.	Міграція кваліфікованих фахівців та членів команди	Членам команди буде запропоновано кращі умови праці, або вони підуть з інших причин	Залучить нових кваліфікованих фахівців у найближчі строки
3.	Скорочення бюджету	Гроші інвесторів перестануть вливатись в проект	Проведення дій для залучення нових інвесторів

Таблиця 5.10 - Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Вихід на ринки	Випуск готового і повноцінного продукту	Підтримка та подальше покращення продукції
2.	Домінування на ринку	Бути єдиною компанією яка поставляє послуги з аналізу системи на проникність у співвідношенні послуги/гроші	Підтримка та подальше покращення продукції

Надалі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку (табл. 5.11).

Таблиця 5.11 - Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1.Вказати тип конкуренції Вільна конкуренція	Немає засобів повноцінного тестування систем на безпеку без постійного втручання людиною	

Продовження таблиці 5.11

2. За рівнем конкурентної боротьби - локальний/ національний/…	У світі немає засобів повноцінного тестування систем на безпеку без постійного втручання людиною	
3. За галузевою ознакою - міжгалузева	Інформаційними системами користуються всі галузі	
4. Конкуренція за видами товарів: - товарно-видова	Іншим варіантом оцінки спроможності оцінки прникності системи до хакерських атак є пентест, який проводиться людиною	Не перевищувати вартість пентестів
5.За характером конкурентних переваг - цінова та нецінова	Пентести є ресурсозатратними, ажде дял їх проведення потрібно задіювати систему, що може на жеякий час зупинити її або пошкодити Також пентести потребують багато коштів	Не перевищувати вартість пентестів

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі за моделлю 5 сил М. Портера (додаток Б) (табл.5.12).

Таблиця 5.12 - Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	Прямими конкурентами є компанії, що проводять тест на проникнення	На даний момент потенційних конкурентів не виявлені	Низький	Високий	Пентести
Висновки:	Більшість людей можуть не довіряти новостворений системі та будуть продовжувати користуватись послугами перевірених компаній	Строками виходу на ринок є півтора роки	Вихід продукту залежить лише від розробників	Від зацікавленості продукцією клієнтами залежить його перебування на ринку	Більшість людей можуть не довіряти новостворений системі та будуть продовжувати користуватись послугами перевірених компаній

Аналізуючи таблиці 5.8-5.12 можна зробити висновок, що можна отримати свою нішу на ринку, якщо пропонувати свої послуги за нижчу ціну, також необхідно надавати результати аналізу інформаційної системи не гірші, ніж отримуються при проведенні тестів на проникнення.

На основі проведеного аналізу конкуренції (табл. 5.12), а також із урахуванням характеристик ідеї проекту (табл. 5.5), вимог споживачів до товару (табл. 5.8) та факторів маркетингового середовища (табл. 5.9-5.10) визначається та обґрунтовується перелік факторів конкурентоспроможності. Аналіз оформлюється за табл. 5.3.10.

Таблиця 5.13 - Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Ціна	Ціна на тестування системи з використанням моделі не потребує залучення великої кількості аналітиків та адміністраторів безпеки.
2.	Менше ресурсів	Не потребує зупинку роботи інформаційної системи та не пошкодить її функціональності.

За визначеними факторами конкурентоспроможності (табл. 5.13) проводиться аналіз сильних та слабких сторін стартап-проекту (табл. 5.14).

Таблиця 5.14 - Порівняльний аналіз сильних та слабких сторін «назва проекту»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з компаніями, що проводять пентести						
			–3	–2	–1	0	+1	+2	+3
1	Ціна							+2	
2	Менше ресурсів								+3

SWOT-аналіз представлений в таблиці 5.15.

Таблиця 5.15 - SWOT- аналіз стартап-проекту

Сильні сторони	Слабкі сторони
Високий рівень кваліфікації фахівців з інформаційних технологій	Короткі строки
Великий бюджет	Аналітика проекту знаходиться на стороні замовника
Великі перспективи залучення нових підпроектних частин в разі успішного виконання	Недостатній нагляд і керівництво
Низька заробітна плата	Реагування на відомі атаки (неуніверсальність)
Наявність досвід у виконанні подібних проектів	
Ентузіазм	
Різноманітність досвіду	
Глибина експертизи	
Можливості	Загрози
Вихід на ринки	Поява конкурентів
Домінування на рику	Перевищення строку виконання
Розширення сфери послуг	Міграція кваліфікованих фахівців та членів команди
Співпраця з іноземними клієнтами та встановлення постійних контактів	Скорочення бюджетів

Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів (табл. 5.16).

Таблиця 5.16 - Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
	Вихід на ринок	Висока	1,5 року

Єдиною альтернативою ринкової поведінки проекту є вихід на ринок.

5.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 5.17).

Таблиця 5.17 - Вибір цільових груп потенційних споживачів

Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
Компанії, що працюють з інформаційними системами	Споживачі готові, бо питання безпеки на сьогодні важливо з законодавчої точки зору	Великий	Недостатня кількість програмного забезпечення з безпеки та недостатня кількість кваліфікованих	Низька

Продовження таблиці 5.17

	токи зору та з економічної точки зору		фахівців з інформаційної безпеки	

Цільовими групами, для яких буде запропоновано продукт, будуть компанії, які хочуть захистити інформацію, яка циркулює в їх інформаційній системі, а це переважно банківські системи, компанії, що працюють з первинними даними, державні структури, де циркулює таємна інформація та інші компанії, в яких циркулює комерційна таємниця.

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (табл. 5.18).

Таблиця 5.18 - Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи
Вихід на ринок	Випустить продукт у продаж	Низька ціна Та невикористання ресурсів системи

Наступним кроком є вибір стратегії конкурентної поведінки (табл. 5.19).

Таблиця 5.19 - Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
	Не є першопроходцем	так	Не буде	Стратегія диференціації

У таблиці 5.20 представлено визначення стратегії позиціонування, що є наступним кроком.

Таблиця 5.20. Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
Максимальна забезпечення безпекою за невисоку ціну	Стратегія диференціації	Низька ціна та невикористання ресурсів системи, що не завдасть їй шкоди	-дешевизна -менше ресурсів

Ринковою поведінкою компанії буде надання послуг безпеки за відносно невисоку ціну.

5.5 Розроблення маркетингової програми стартап-проекту

Першим кроком розроблення маркетингової програми стартап-проекту є визначення ключових переваг концепції потенційного товару.

Таблиця 5.21 - Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
1.	Нижча ціна	Нижчу ціну у відношенні до конкурентів	Нижча ціна
2.	Відсутність збоїв у роботі системи	Не потребує залучення самої сстеми	Не потребує ресурсів системи
3.	Забезпечення безпеки	Забезпечує аналіз на безпечність системи	Забезпечує не гуршу якість

Наступним кроком є опис трьох рівнів моделі товару, що представлені в таблиці 5.22.

Таблиця 5.22 - Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за задумом	Забезпечення за невисоку ціну аналіз системи на проникність від хакерських атак без залучення самої системи

Продовження таблиці 5.22

II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Аналіз системи на проникність		
	Якість: ліцензована та сертифікована компанія за міжнароднними стандартами		
	BMS consalting		
III. Товар із підкріпленням	До продажу – тестування на проникність проводиться з безпосереднім залученням самої системи		
	Після продажу – модель системи дозволить не експлатувати для тестування систему.		
За рахунок чого потенційний товар буде захищено від копіювання: Для захисту ПЗ від копіювання буде застосовано механізм ліцензування.			

Захист інтелектуальної власності буде організовано за рахунок комплексному поєднанню властивостей і характеристик товару.

Таблиця 5.23 - Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
	Починається від $2000	Починається від $2000	Від $4000	$2000 - $6000

В якості основного товару замінника було обрано тести на проникнення, а їх ціни будо взято з інтернет ресурсу .

Після визначення цін формується системи збуту товару, що представлена в таблиці 5.23.

Таблиця 5.23 - Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
	Власна	Встановлення та супроводження ПЗ	Канал нульового рівня	Встановлення та супроводження ПЗ клієнту

Останні кроком є визначення концепції маркетингових комунікацій, адже цільовим завданням створення проекту є його успіх (табл. 5.24).

Таблиця 5.24 - Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Надання індивідуального підходу	Безпосереднє спілкування	Ціновий показник та залучення ресурсів	Зацікавити якомога більшу кількість компаній у встановлення розробленого ПЗ	Розкрити основні плюси та особливості продукту

Компанії в яких циркулює таємна, комерційна інформація або персональні дані прикладають зусилля для захисту цієї інформації, тому допомога у використанні

програмного забезпечення та його подальша підтримка надає можливість клієнту впевнитись, що воно застосовуються правильно і задіяний його повний функціонал.

Висновки до розділу 5

Отже, розробка моделі для оцінки спроможності проникнення хакером та спроможності їм протидіяти є надзвичайно актуальною, бо велика кількість важливої та коштовної інформації циркулює саме в кібернетичному просторі. Але на сьогодні проблема безпеки є критичною через дороговизну залучення аналітиків з інформаційної безпеки та їх недостатню кваліфікацію. Створення системи, що зможе моделювати атаку та впроваджені системи захисту спросить задачу захисту через відсутність залучати коштовних спеціалістів та на деякий час зупиняти систему для її тестування, що може економічно зашкодити компанії та її репутації.

Враховуючи все вище перечислене ідеєю розробки стало пілкомплексне поєднання властивостей і характеристик.

Обраною технологією стали мережі Петрі у зв’язці з мовою програмування Java. Причиною такого висновку стали відсутність програмного забезпечення, яке варте було б дорозробки, тому створення власного фреймворку для мереж Петрі на Java, з допомогою якого та JavaFX для візуалізації буде створено модель хакерських атак на інформаційну систему.

Також було визначено та зроблено висновок, що даний проект є високорентабельним. Оскільки в перший рік його реалізації завершувались наукові дослідження та освоювалось обладнання та навчався персонал роботі на даному обладнанні, то грошові потоки в перший рік забезпечувались у вигляді вливання сторонніх інвестицій. Реалізація перших замовлень розпочинається тільки з другого року . За два з половиною роки компанія зможе розрахуватись з інвестором, а наступні роки буде працювати на власний прибуток. Термін окупності даного

проекту визначається періодом між другим та третім роками (коли ЧПВ > 0, то проект окупається).

Цільовою аудиторією є компанії в яких циркулює таємна, комерційна інформація або персональні дані прикладають зусилля для захисту цієї інформації, тому допомога у використанні програмного забезпечення та його подальша підтримка надає можливість клієнту впевнитись, що воно застосовуються правильно і задіяний його повний функціонал.

ВИСНОВКИ

Для досягнення мети було виконані наступні завдання:

a) Досліджено умови виконання інформаційних атак. Визначено, що життєвий цикл атаки складається з трьох етапів: збору інформації, здійснення нападу і поширення атаки.

b) Досліджено загальну структуру інформаційної системи та визначено її основні вразливості. Повноцінне дослідження структури неможливе без втручання в саму інформаційну систему, можливо дослідити лише хости, які мають доступ до Інтернету. Також пасивне визначення вразливості можливе за умови відкритого порту у сервісу та за можливості визначити його версію. Після визначення вище перерахованих сигнатур вразливостей, інформація про саму вразливість може бути взята з публічних агрегаторів інформації про них: численні бази даних (NVD, VulnDB та інші.) та відкриті інформаційні ресурси.

c) Розроблено моделі атак та захисту інформаційної системи за допомогою мереж Петрі, шляхом попереднього збору інформації про комп'ютерну систему, сервісів та наявних вразливостей. Основну логіку моделі формують характеристики вразливостей, а саме:

- вектор враження;
- складність експлуатації вразливості;
- права для експлуатації вразливості;
- права, які отримує зловмисник після експлуатації вразливості.

d) Розроблено застосунок для моделювання хакерських атак на систему, що складається з двох модулів написаних мовою програмування Java, перший – це бібліотека для об'єктно-орієнтованого представлення Петрі-об'єктів, а друга для збору інформації про систему та вразливості.

е) Виконано експериментальне дослідження розробленої моделі та отримано числові значення і вигляді відносного часу здійснення хакером атаки певним шляхом для досягнення своєї цілі. Ці дані дають дають можливість адміністратору безпеки визначити критичні точки в системі та зробити висновки щодо їх усунення або прийняття ризиків.

Використання даного методу дозволяє створити моделі, на основі простої конструкції для представлення різних типових дій. Завдяки використанню мереж Петрі, метод дозволяє створювати поточний стан ІС і об'єкт атаки під час моделювання – забезпечує динамічне моделювання. Крім того, моделі Петрі дозволяють використовувати математичний апарат із описом параметрів моделей, створюють формальну сутність.

Застосування такого підходу дозволяє вирішити проблеми багатофакторності (комплексності) атаки, виділення етапу пошуку управляючих об'єктів атаки (рівень планування), проведення реалізацій атаки та визначення її можливих останніх (реактивний рівень); для обміну інформацією між агентами різних рівнів існує фактичний кооперативний рівень.

Отримані імітаційні моделі інформаційних дій можуть бути використані для побудови синтетичних округлих системних інформаційних систем безпеки з цілком уточненими їх особливостями та характеристиками з підтримкою методів напівнатуральних моделей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) Wolfgang Reisig. Petri Nets: An Introduction. Monographs in Theoretical ComputerScience. Springer Verlag, Heidelberg, Germany, 1985.

2) Matt Bishop and Michael Dilger. Checking for Race Conditions in File Accesses.Computing Systems, 9(2):131–152, 1996

3) James Lyle Peterson. Petri Net Theory and the Modeling of Systems. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1981.

4) Kurt Jensen. Coloured Petri Nets: Basic Concepts (Volume 1). Monographs in Theoretical Computer Science. Springer Verlag, Heidelberg, Germany, 1997.

5) Carl A. Petri. Kommunikation mit Automaten. PhD thesis, Technische Hochschule Darmstadt, Darmstadt, Germany, 1962. Published in Schriften des Instituts für Instrumentelle Mathematik 3,1-128, University of Bonn, Germany.

6) Tina Darmohray. hacking for fun and profit. ;login, April 2003.

7) Методології розробки програмного забезпечення. – [Електронний ресурс]. – Режим доступу: http://habrahabr.ru/sandbox/43802/

8) Алексенко О. В. Технології програмування та створення програмних продуктів. – [Електронний ресурс]. – Режим доступу: http://essuir.sumdu.edu.ua/bitstream/123456789/30254/1/Alekseenko_Programuvannja.pdf

9) Моделі життєвого циклу. – [Електронний ресурс]. – Режим доступу:http://www.informicus.ru/default.aspx?id=73&SECTION=6&subdivisionid=7

10) Microsoft® Operations Framework. – [Електронний ресурс]. – Режим доступу: http://wikiitil.ru/books/MOF-1-Overview(rus).pdf

11) ДСТУ 2844-94 Програмні засоби ЕОМ. Забезпечення якості. Терміни та визначення. – [Електронний ресурс]. – Режим доступу:

12) .Державні стандарти по захищеності інформаційних систем – [Електронний ресурс]. – Режим доступу:http://www.csm.kiev.ua/nd/nd.php?b=1&l=4017

13) Tim Dierks and Christopher Allen. The TLS Protocol Version 1.0. RFC 2246 (Proposed Standard), January 1999. Updated by RFC 3546.

14) Martin Freiss. Protecting Networks with Satan. O'Reilly Press, 1997.

15) Daniel Geer and John Harthorne. Penetration Testing: A Duet. In Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC 2002), pages 185–198, Las Vegas, NV, USA, December 2002. IEEE Press.

16) Frank Swiderski and Window Snyder. Threat Modeling. Microsoft Press, 2004.

17) .Тестування на проникнення – [Електронний ресурс]. – Режим доступу:https://www.itu.int/en/ITU-D/Regional-Presence /CIS/Documents/Events /2018/ 04_Odessa/Presentations/ITU%20Workshop20Apr2018- OlehDubina.pdf

18) George Kurtz and Chris Prosise. Penetration Testing Exposed. Information Security Magazine, September http://www.infosecuritymag.com/ Visited May 2005.

19) Shodan . – [Електронний ресурс]. – Режим доступу:https://www.sho dan.io

20) Censys: – [Електронний ресурс]. – Режим доступу:https://censys.io/

21) National Vulnerability Database . – [Електронний ресурс]. – Режим доступу:https://nvd.nist.gov

22) Stetsenko I.V., Dyfuchyn A., Leshchenko K., Davies J.: Web application for visual modeling of discrete event systems. In: Picking R., Cunningham S., Houlden N., Oram D., Grout V., Mayers J. (eds) Proceedings of the Seventh International Conference on Internet Technologies and Applications (ITA2017), pp. 86-91, Wrexham, UK (2017).

23) Common Vulnerabilities and Exposures. – [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Common_Vulnerabilities_and_Exposures, last accessed 5 November 2019.

24) CVE official. – [Електронний ресурс]. – Режим доступу:https://cve.mitre.org

25) CVE format. – [Електронний ресурс]. – Режим доступу: https://nvd.nist.gov/vuln/detail/CVE

26) Top 5 web servers. – [Електронний ресурс]. – Режим доступу: https://binge.co/what-are-the-best-web-servers

27) Shodan REST API Documentation. – [Електронний ресурс]. – Режим доступу: https://developer.shodan.io/api/

28) Shodan client for Java. – [Електронний ресурс]. – Режим доступу: https://github.com/fooock/jshodan/

29) Common Vulnerability Scoring System terms https://en.wikipedia.org/wiki/Common_Vulnerability_Scoring_System#cite_note-cvss3.1-1/

30) CVE Details – the ultimate security datasource. – [Електронний ресурс]. – Режим доступу: https://www.cvedetails.com

31) Common Vulnerability Scoring System. – [Електронний ресурс]. – Режим доступу:https://www.first.org/cvss//

32) Common Weakness Enumeration. – [Електронний ресурс]. – Режим доступу:https://cwe.mitre.org/index.html/

33) Selenium - Web Browser Automation. – [Електронний ресурс]. – Режим доступу: https://www.seleniumhq.org//

34) Shodan REST API Documentation . – [Електронний ресурс]. – Режим доступу: https://developer.shodan.io/api/

35) Shodan client for Java. – [Електронний ресурс]. – Режим доступу: https://github.com/fooock/jshodan/

36) ORM Hibernate. – [Електронний ресурс]. – Режим доступу: https://hibernate.org/

37) jShodan. – [Електронний ресурс]. – Режим доступу: https://github.com/fooock/jshodan

38) Spring data. – [Електронний ресурс]. – Режим доступу: https://spring.io/projects/spring-data

39) PostgreSQL. . – [Електронний ресурс]. – Режим доступу :https://www.postgresql.org/

40) Лукацкий А.В. Обнаружение атак / Лукацкий А.В. – СПб.: БВХ-Петербург, 2001. – 624 с.

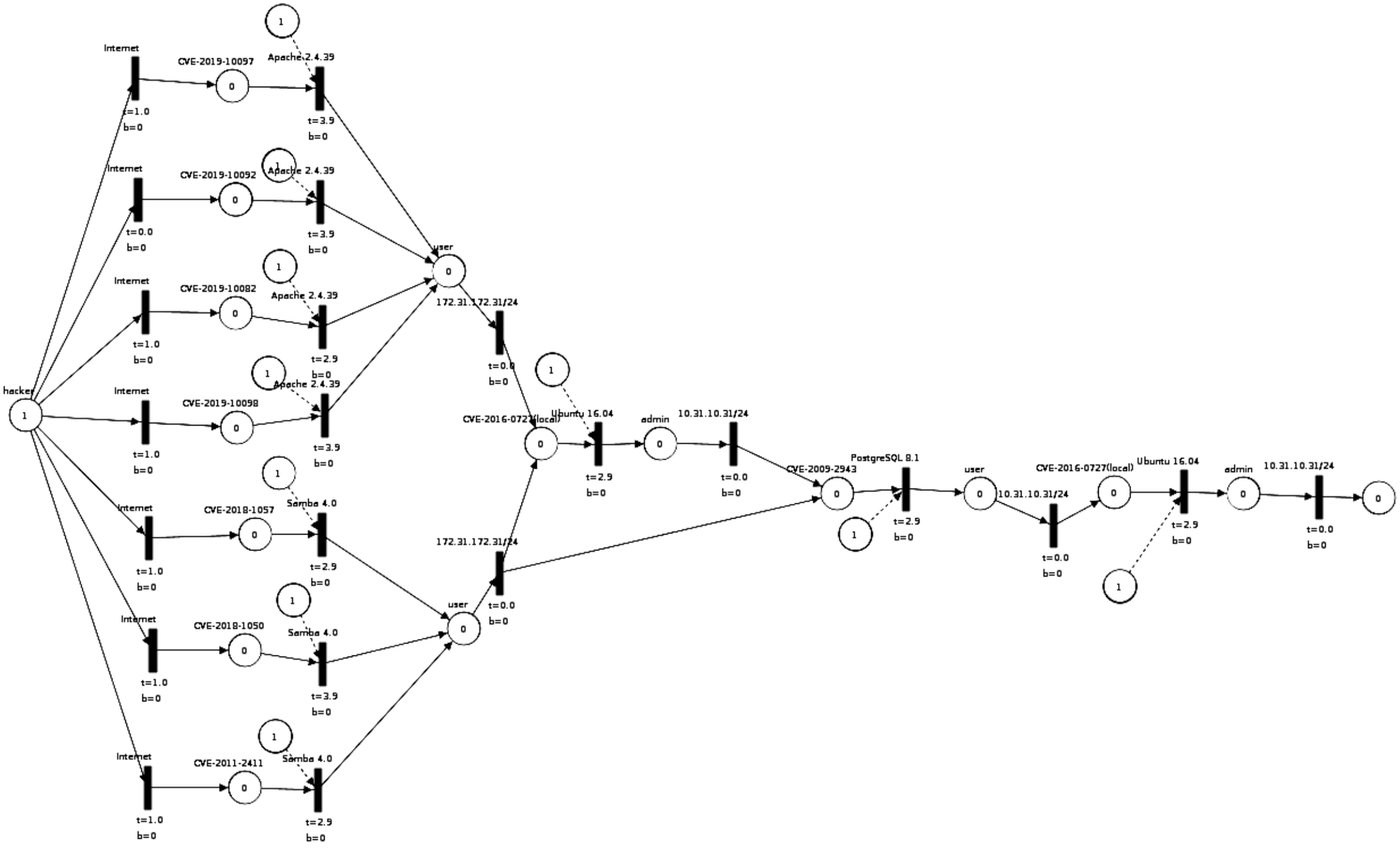
41) Сердюк В.А. Информационная безопасность автоматизированных систем предприятий /В.А. Сердюк // Бухгалтер и компьютер. – 2007. – No 1. – С. 104 – 107.

42) Натров В.В. Классификация сетевых атак / В.В. Натров // Информационные технологии в управлении и моделировании: сб. докладов. – Белгород, 2005. – С. 128 – 132.

43) Котенко И.В. Моделирование противоборства программных агентов в Интернете: общий подход, среда моделирования и эксперименты / И.В. Котенко, А.В. Уланов // Защита информации. IN-SIDE. – 2006. – No 5. – С. 2 – 10.

ДОДАТОК А
ГРАФІЧНІ ЗОБРАЖЕННЯ

МОДЕЛЬ ПОШИРЕННЯ АТАК МЕРЕЖЕЮ ПЕТРІ



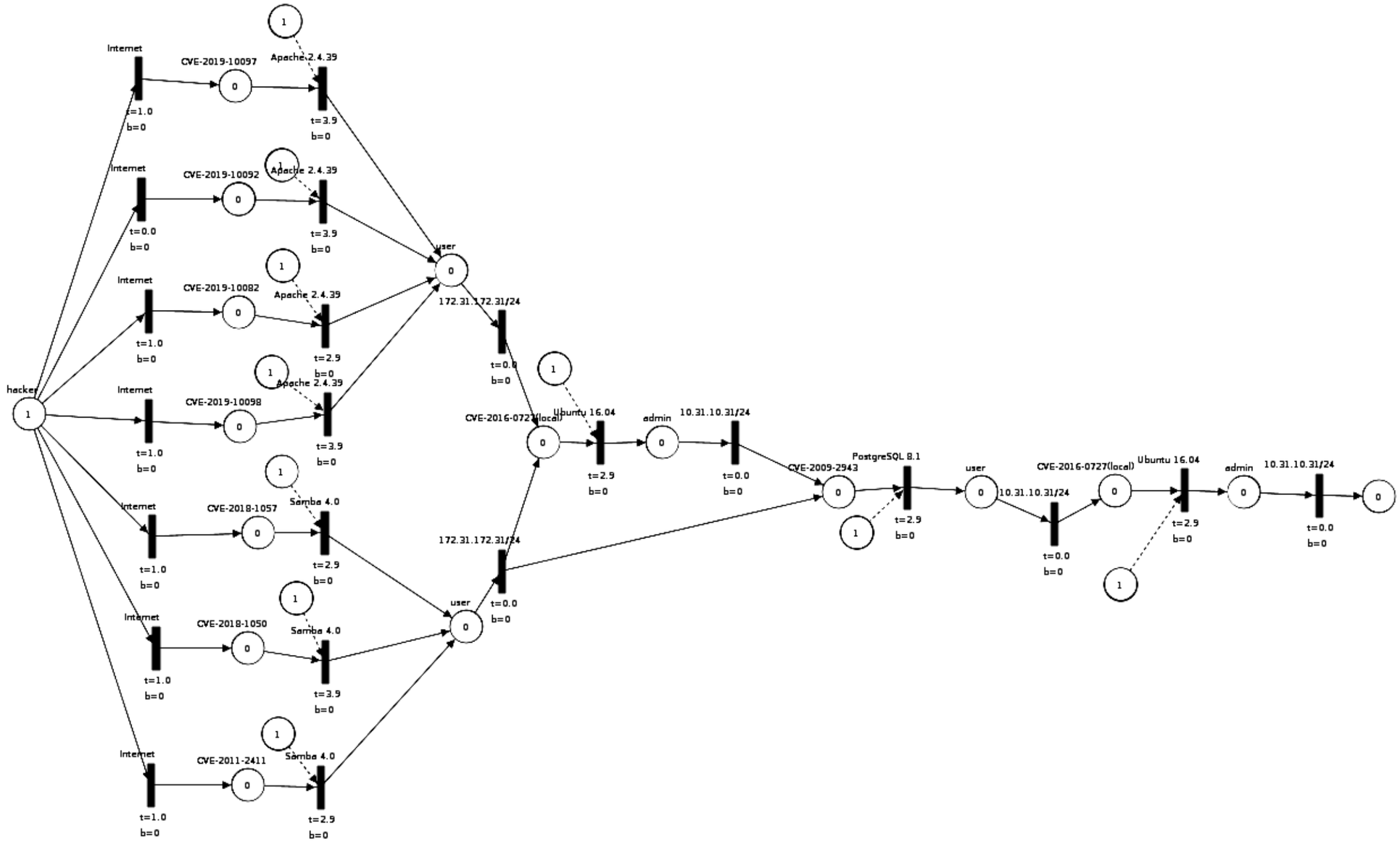
Демонстраційний плакат до магістерської дисертації

Математичне та програмне забезпечення оцінювання захищеності інформаційних систем від веб-атак

Виконала студентка гр. ІП-82мп Савчук В.В.

Керівник Стеценко І.В.

МОДИФІКОВАНА МОДЕЛЬ ПОШИРЕННЯ АТАК МЕРЕЖЕЮ ПЕТРІ



Демонстраційний плакат до магістерської дисертації

Математичне та програмне забезпечення оцінювання захищеності інформаційних систем від веб-атак

Виконала студентка гр. ІП-82мп Савчук В.В.

Керівник Стеценко І.В.

ДОДАТОК Б

МОДЕЛЬ АНАЛІЗУ КОНКУРЕНЦІЇ У ГАЛУЗІ М. ПОРТЕРА

М. Портер вирізняє п’ять основних факторів, що впливають на привабливість вибору ринку з огляду на характер конкуренції. Це:

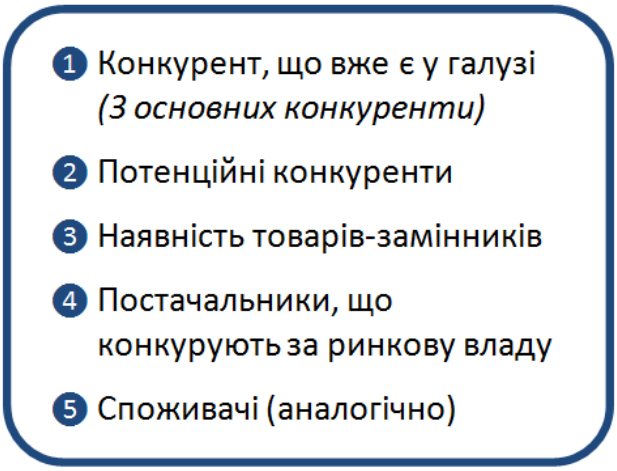


Рисунок Б.1 - Складові моделі 5 сил М. Портера

Сильні позиції компанії за кожним з факторів означають її можливості забезпечити необхідні темпи обороту капіталу та її здатність впливати на інших агентів ринку, диктуючі їм власні умови співпраці. Характеристики факторів моделі відрізняються для різних галузей та змінюються із часом. Сила кожного фактору є функцією від структури галузі та її техніко-економічних характеристик.

На основі аналізу складових моделі 5 сил М. Портера розробляється перелік факторів конкурентоспроможності для певного ринку.